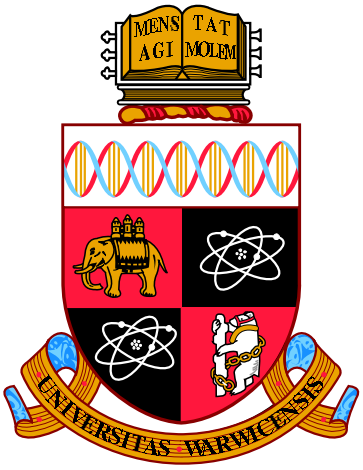


Coupling Methods for 2D/1D Shallow Water Flow Models for Flood Simulations

by

Chinedu Nwaigwe

Thesis

Submitted to the University of Warwick

for the degree of

Doctor of Philosophy

Centre for Scientific Computing

2016

THE UNIVERSITY OF WARWICK

Contents

List of Tables **v**

List of Figures **vi**

Acknowledgments **xii**

Abstract **xiii**

Abbreviations **xiv**

Chapter 1 Introduction **1**

1.1 Introduction 1
1.2 Flood Flow Simulation 3
1.3 The Main Ideas in Coupling Channel and Floodplain Flow Models . 4
1.4 Research Objectives 6
1.5 Review of Coupling Methods 8
1.6 Outline of the Thesis 12

Chapter 2 Mathematical Models for Flood and Channel Flows **14**

2.1 Introduction 14
2.2 The Free-Surface Euler Equations 15
2.2.1 Background 15
2.2.2 Deriving the free-surface Euler Equations 16
2.3 Derivation of 2D Shallow Water Equations 18
2.4 1D Saint Venant Equations for an Open Channel With Varying Width 21
2.4.1 Background 22
2.4.2 Deriving the SVM 24
2.4.3 Locally Rectangular Channel 29
2.4.4 1D Shallow Water Equations 29
2.5 Properties of Hyperbolic Conservation Laws 29

2.5.1 Hyperbolic PDEs 30
2.5.2 Hyperbolic Conservation Laws 30
2.5.3 Hyperbolicity of the Derived Models 32
2.5.4 Properties of hyperbolic PDEs 33

Chapter 3 Finite Volume Schemes For Free-Surface Flow Problems 36
3.1 Introduction 36
3.2 Conservative Numerical Schemes 38
3.3 Finite Volume Methods for Conservation Laws 40
3.4 Finite Volume Method for the Flood Model 41
3.4.1 The 2D Shallow Water Equations without Source Terms . . . 41
3.4.2 The 2D Shallow Water Equations with Bottom Topography . 44
3.4.3 The 2D Shallow Water Equations with Bottom Topography and Friction 46
3.5 Finite Volume Method for the Channel Flow Model 47
3.6 Boundary Conditions 50
3.6.1 Solid Boundary 51
3.6.2 Open Boundary 51
3.7 Time step calculations 52
3.8 Numerical Results 53
3.8.1 Test Cases for the 1D Channel Flow Solver 53
3.8.2 Test Cases For the 2D Floodplain Flow Solver 57
3.8.3 Why Coupling is needed 63

Chapter 4 The Horizontal Coupling Method 68
4.1 Introduction 68
4.2 The Channel Model with Coupling Terms 69
4.2.1 Notational Simplification 74
4.2.2 Lateral Discharge Model During Flooding 77
4.2.3 Summary of the Channel Flow Model for the HCM 77
4.3 Floodplain Flow Model 78
4.4 Summary of all Models in the HCM 78
4.5 Numerical Implementation of the HCM 78
4.5.1 Numerical Scheme for the Channel Flow Model 79
4.5.2 1D SVM with Coupling Term 80
4.5.3 Approximating Channel Lateral Discharge 81
4.5.4 Discrete Coupling Terms 84
4.5.5 Summary of the Schemes for the Channel Flow Model 86

4.5.6 Summary of the Schemes for the Horizontal Coupling Method 87
4.6 Properties of the HCM 87
4.7 Implementation Strategy 89
4.7.1 Black Box Solver stage 90
4.7.2 Post-Processing Stage 90
4.8 Numerical Results 91

Chapter 5 The Vertical Coupling Methods (VCM) 95
5.1 Introduction 95
5.2 Background 97
5.2.1 Fundamental Equations 102
5.2.2 Defining the layers 103
5.3 The Lower Layer Models 105
5.3.1 Mass Conservation Equation for Lower Layer 105
5.3.2 Momentum Equation for Lower Layer 109
5.3.3 Summary of Lower layer Models 110
5.4 The Upper Layer Models 110
5.5 Summary of the Channel Flow Model in the VCM 111
5.6 Numerical Implementation of the VCM 113
5.6.1 Introduction 113
5.6.2 Background 114
5.6.3 Step 1: Distribution of Full 2D data into two-layer data . . . 120
5.6.4 Step 2: Operator Splitting Method For Sub-Layer Models . . 121
5.6.5 Step 3: Updating full $2D$ data $H_{i,j}^{n+1}$, $q_{x,i,j}^{n+1}$ and $q_{y,i,j}^{n+1}$ for $T_{i,j} \in K_i$ 126
5.6.6 Finite Volume Methods for the intermediate solutions 129
5.6.7 Solution of the Upper Layer Model without Exchange Terms 129
5.6.8 Solution of the Lower Layer Model without Exchange terms 130
5.7 Desirable Properties of the VCM Scheme 131

Chapter 6 Numerical Results for Coupling Methods 139
6.1 Introduction 139
6.2 Test case 1 141
6.3 Test case 2 : Channel Flow into Elevated 2D Floodplain 142
6.4 Test case 3 : Flooding of an initially dry floodplain 146

Chapter 7 Summary, Conclusion and Recommendations 158
7.1 Summary 158
7.2 Conclusion 159

7.3 Recommendation for further study 160

Appendix A Numerical Flux Functions **162**

A.1 Godunov-Type Finite Volume Schemes 162

A.2 Approximate Riemann Solvers 163

A.2.1 Roe Solver 163

A.2.2 HLL Approximate Riemann Solver 168

A.2.3 HLLC Approximate Riemann Solver 169

A.3 Centered Numerical Flux Solvers 170

List of Tables

3.1 Experimental Order of Convergence for the 1D Riemann Problem (1D test case 1). E_A, E_Q are the errors in the cross sectional area, A and discharge, Q respectively, while EOC_A and EOC_Q are their respective orders of accuracy. 56

3.2 Experimental Order of convergence for the 2D scheme using the flow in parabolic basin problem (2D test 2). E_H, E_{q_x} and E_{q_y} are the errors in l_1-norm of H, q_x and q_y respectively, while EOC_H, EOC_{q_x} and EOC_{q_y} are their respective experimental order of convergence. . 61

4.1 Grid cells, simulation times and number of time steps after ten seconds. 91

6.1 Grid cells, simulation times and number of time steps for Test 1. Shows the grid used for each simulation method and the number of time steps and processor time taken to complete the simulation. . . 142

6.2 Grid cells, simulation times and number of time steps : Test 2. Shows the grid used for each simulation method and the number of time steps and processor time taken to complete the simulation. 146

6.3 Grid cells for test case 3 showing the grids used for the floodplains and the channel for each simulation method. 153

List of Figures

1.1 Shallow water flow over a domain with bottom topography $z_b(x, y)$ (in red) comprising of a channel and floodplain. The length of the channel is along the x-axis and the width, along the y-axis. The water depth is $H(x, y, t)$ and the free surface elevation is $\eta(x, y, t)$ (in green). Symbol t is time variable and $(x, y, z) \in \mathbb{R}^3$. 4

1.2 Cross section at a fixed point x in the flow domain depicted in figure 1.1. The bottom topography (in red) comprises of the channel in the region $y_a \leq y \leq y_b$ and the floodplain which occupies the remaining regions. The free surface elevation is in green and $H(x, y, t)$ is the water depth. 5

1.3 Cross section at a fixed x in the flow domain showing the different regions for channel and the floodplains. The middle region is the channel which occupies the region $y_a \leq y \leq y_b$ while the left and right regions are the floodplains. 5

1.4 A picture of the different models for the different regions. The 2D shallow water equations for the floodplain flow (left and right) and a channel flow model for the channel region (middle). 6

1.5 Flow cross section depicting only the channel region (channel cross section). Channel model is derived for the flow in this region. The flow in the floodplains are used as boundary conditions. The channel flow also serve as boundary conditions for the 2D floodplain models. 7

1.6 Channel flow structure for the horizontal coupling method (left) and the vertical coupling method (right). 7

2.1 Shallow water flow over a domain with bottom topography $z_b(x, y)$ (in red). The water depth is $H(x, y, t)$ and the free surface elevation is $\eta(x, y, t)$ (in green). Symbol t is time variable and $(x, y, z) \in \mathbb{R}^3$. . 16

2.2 Non-full channel flow cross section showing the left and right bank elevations $z_{bl}(x)$ and $z_{br}(x)$, the laterally flat free-surface elevation $\bar{\eta}(x,t)$ (in green), the bottom elevation in 2D $z_b(\vec{\boldsymbol{X}})$ (in red), the width, $B(x,z)$ of the channel at elevation, z above the reference elevation $z=0$, and the $y-$coordinates, $y_l(x,z)$ and $y_r(x,z)$ of the left and right lateral walls at elevation z. 22

3.1 Discretization of the 1D domain $[x_a, x_b] \subset \mathbb{R}$ into $N+1$ grid cells. $K_i = [x_{i-1/2}, x_{i+1/2}]$ is the i-th cell, for $i = 0, 1, 2, ..., N$. $x_i = \frac{x_{i-1/2}+x_{i+1/2}}{2}$ and $\Delta x_i = x_{i+1/2} - x_{i-1/2}$ are center and width of K_i respectively. 37

3.2 2D mesh showing two neighbour cells, T_j and T_k, the edge e_{jk} between them and the normal vector $\vec{n}_{jk}$. 42

3.3 Exact and Numerical Results for the Riemann problem showing the free surface elevation (left column) and velocity (right column) at different time steps. From the top to the bottom are the results at the 100th, 200th, 400th and the last (465th) time steps. In each plot, the exact and numerical solutions are in blue and magenta respectively. 55

3.4 The channel width and bed variations for the 1D flow in varying geometry channel. 57

3.5 Numerical Solution for the flow in channel with varying geometry (1D test 2). 57

3.6 Bed elevation and initial free surface elevation for still water 58

3.7 Numerical Results after 5 seconds for the still water over complex bottom topography 59

3.8 The bottom elevation and the initial water depth and discharges for 2D test case 2. 61

3.9 Numerical (left) and analytical (right) results for the water height and the discharges after 5.93018 seconds for 2D test case 2. 62

3.10 Top view of Channel and Floodplain for river-flooding problem . . . 63

3.11 Final Free Surface Elevation, η after ten seconds. 64

3.12 Comparison of the simulation results with the experimental data for the free surface elevation, η after the last time step. The positions of the probe points are indicated in figure 3.10. 65

3.13 Lateral Variation of the flow quantities within the channel after ten seconds (the last time step). 67

4.1 Top view of 2D flow domain, $\Omega^2 = \Omega_c \cup \Omega_f$ consisting of the river channel, Ω_c and the floodplains, $\Omega_f = \Omega_{f_1} \cup \Omega_{f_2}$. 70
4.2 Cross sectional view of 2D flow domain, $\Omega^2 = \Omega_c \cup \Omega_f$ consisting of the river channel, Ω_c and the floodplains, $\Omega_f = \Omega_{f_1} \cup \Omega_{f_2}$. 70
4.3 Channel cross section in HCM without the floodplains, showing the channel bottom topography, $z_b(x, y)$ (in red), the channel wall elevation , $z_b^w(x)$, laterally flat free-surface elevation $\bar{\eta}(x, t)$, the bottom elevation in 1D sense $Z_b(x)$, the top width $B(x, z_b^w(x))$ and the $y-$coordinates $y_l^w(x) := y_l(x, z_b^w(x))$ and $y_r^w(x) := y_r(x, z_b^w(x))$ respectively of the left and right lateral walls at the channel top. $H(x, y, t)$ is the depth of water measured from $z_b(x, y)$ to the flat free surface $\bar{\eta}(x, t)$. 71
4.4 Top view of Lateral Boundaries (at elevation, $z = z_b^w(x)$) 74
4.5 Diagram explaining the equations solved in the channel. The equations are the 1D Saint Venant model with coupling terms (4.31) and (4.32) and the lateral discharge equation (4.34). Dotted lines indicate the end of channel region at which the lateral fluxes between the channel and floodplains are computed. 77
4.6 Summary of the different models for the different regions of the flow domain. The 1D Saint Venant model with coupling terms, (4.31) and (4.32) and the lateral discharge equation (4.34) are solved for the channel flow while the 2D shallow water system (2.32) is solved for the flow in the floodplains. The dotted lines indicate the boundaries between the sub domains, the point at which lateral fluxes are computed and the blue arrows indicate the presence of flow exchange between them. 78
4.7 Grid of the entire domain consisting of the 1D grid Ω_h^{1D} at the middle and the 2D grids Ω_h^{2D} for the floodplains. The grids are matching in the sense that there is no gap between the 1D and the 2D grids. . . 79
4.8 A single cell, K_i in the 1D channel mesh showing its lateral edges; South edge e_i^S is on the negative y-direction while the North edge e_i^N is on the positive y-direction. These edges are the interfaces between the 1D cell and the adjacent 2D floodplain cells. 80
4.9 To the left is a 1D channel cell and its adjacent 2D floodplain cells while to the right is the 1D cell subdivided into two subcells viewed as 2D cells 82

4.10 Flow Chart for horizontal Coupling Method depicting the black box solver and post processing stages. The black box solver stage computes $\boldsymbol{W}_i^{n+1*}$ using the 1D solver (3.38) and Π_{ij}^{n+1}, with the 2D solver (3.30). At the post processing stage, Φ_i^n is computed with (4.60), while $(q_y^{S/N})_i^{n+1}$ are computed using (4.46) and (4.47). 93
4.11 Comparison of the final free surface elevation for the HCM with those of the full 2D simulation and the FBM after ten seconds. 94
4.12 Comparison of the time evolution of the free surface elevation, η at the probe points indicated in figure 3.10. 94

5.1 Cross sectional view of 2D flow domain, $\Omega^2 = \Omega_c \cup \Omega_f$ consisting of the river channel, Ω_c and the floodplains, $\Omega_f = \Omega_{f_1} \cup \Omega_{f_2}$. The green curve is the free surface elevation and the red curve is the bottom topography. 97
5.2 Cross section of the channel flow domain without the floodplains in the VCM. It shows the channel bottom topography $z_b(x, y)$ (in red), the channel wall elevation $z_b^w(x)$, laterally varying free-surface elevation $\eta(x, y, t)$, the bottom elevation in 1D sense $Z_b(x)$, the top width $B(x, z_b^w(x))$ and the $y-$coordinates $y_l^w(x)$ and $y_r^w(x)$ of the left and right lateral walls at the channel top. The total water depth $H(x, y, t)$, the channel depth $\beta(x, y)$, the lower layer water depth $h_1(x, y, t)$ are all measured from the same bottom elevation, $z_b(x, y)$. The upper layer water depth is $h_2(x, y, t)$. 98
5.3 The two layers in the VCM in the case of full channel (left) and the non full case (right). 104
5.4 Mesh for the two sub-models. 114
5.5 Three dimensional view of the channel meshes (without floodplain) showing a single 1D channel cell, K_i (the largest rectangle in blue) and the 2D channel cells, T_{ij} (the smaller blue rectangles). The channel bottom topography is shown in black. The arrows indicate the flow exchange between the two layers in the channel and it is computed during the second stage of the operator splitting method presented later. 114

5.6 Three dimensional view of a single 1D channel cell, K_i (the largest rectangle in blue), the 2D channel cells, T_{ij} (the smaller blue rectangles) and the floodplain mesh (in green). The arrows indicate the flow exchange between the channel and the floodplain. These interactions are seamlessly computed by the 2D upper layer model while computing the intermediate solutions in the channel. The two layer model is not solved in the floodplain (green) mesh. 115
5.7 Discrete channel geometry in 1D channel cell (figure 5.7(a)) and the cross sectional view of the channel flow 2D channel cells when full (figure 5.7(b)) and non-full (figure 5.7(c)). 116
5.8 Flow chart for the VCM depicting the floodplain flow solver and the channel flow solver. The details of the channel solver is illustrated in figure 5.9. 127
5.9 Details of the channel solver part in the flow chart of the VCM (figure 5.8). It depicts the three-step channel flow solver which implements the two-layer coupled models, (5.46), (5.47). It also shows the two stages to solve (step 2) the channel models. Step 2, stage 1 computes the intermediate solutions and the lateral fluxes between the channel and the floodplains, while step 2, stage 2 enforces the mass and momentum conservation to update each layer solution. 128

6.1 Comparison of free surface elevation for the different methods for test case 1 after the last time step. 143
6.2 Comparison of free surface elevation, η at probe points for test case 1. The locations of the probe points are shown in 3.10. 144
6.3 Top view of Channel and Floodplain for test case 2 showing the floodplain region in $(x, y) \in [10.5, 16.0] \times [0, 1.8]$ and the channel region in $(x, y) \in [0, 19.3] \times [1.8, 2.3]$. 145
6.4 Comparison of free surface elevation distribution for the different methods for test case 2 at the last time step. 147
6.5 Comparison of velocity magnitude for the different methods for test case 2 after the last time step. 148
6.6 Comparison of the time variation of the water height at the probe points for test case 2. 149
6.7 Comparison of time variation of the x-velocity component at the probe points for test case 2. 150

6.8 Comparison of the time variation of the y-velocity component at probe the points for test case 2. 151
6.9 The bottom topography and the channel wall elevation for test 3 . . 152
6.10 Visualisation of free surface elevation after $t = 40$ for test case 3. The x-axis is from left to right, while the y-axis is from the bottom to the top. 153
6.11 Visualisation of x-velocity after $t = 40s$ for test case 3. The x-axis is from left to right, while the y-axis is from the bottom to the top . . 154
6.12 Visualisation of y-velocity after $t = 40s$ for test case 3. The x-axis is from left to right, while the y-axis is from the bottom to the top. . . 154
6.13 Visualisation of velocity magnitude after $t = 40s$ for test case 3. The x-axis is from left to right, while the y-axis is from the bottom to the top 155
6.14 Time variation of water depth H (left column), x-velocity component (middle column) and y-velocity component (right column) at selected probe points within the channel for test case 3. Each row corresponds to one probe point which is indicated on the y-axis of the water depth H plot on the left. 156
6.15 Time variation of water depth H (left column), x-velocity component (middle column) and y-velocity component (right column) at the indicated probe points in the floodplain for test case 3. Each row corresponds to one probe point which is indicated on the y-axis of the water depth H plot on the left. 157

Acknowledgments

Firstly, I would like to express my sincere gratitude to my supervisor Dr Andreas Dedner for his continuous support of my Ph.D study. Dr Andreas' vast knowledge and expertise were not only source of ideas but have also been great motivation to me throughout the period. Every meeting with him during the period provided me with the relevant answers that I needed and initiated great thoughts and reflections that broadened the scope of this study. His patience, especially during the tough times in the Ph.D pursuit, is deeply appreciated.

I am grateful to the Petroleum Technology Development Fund (PTDF), Nigeria for funding this study. I thank the former President of Nigeria Dr Goodluck Jonathan who provided Nigerians equal opportunities devoid of godfather-ism, hence people like me could compete and obtain this prestigious scholarship award.

I also wish to thank the Director of the Center for Scientific Computing, Prof. Mark Rodger for his kindness towards me before and throughout the period of my study in Warwick. Prof. Rodger's support was instrumental for my being a recipient of the Chancellor's International Scholarship award, even though I later turned it down for the PTDF award. I also express my appreciation to all the staff in Warwick, especially Mrs. Glanville Vida, the CSC secretary who has been more than supportive. To my follow Ph.D students, I say thanks for the times we shared together especially during lunch and FATNode dinners.

Special thanks to my family: my wife, my sons, my mother and to my brothers and sisters for their continuous prayers.

Finally, I return all glory to my heavenly father, the Almighty God for making this Ph.D journey a success.

Abstract

Efficient methods for the numerical simulation of the shallow water equations are important for understanding flooding events and related phenomena. One such approach is to couple 2D shallow water floodplain flow model to 1D Saint Venant's open channel flow model. Currently, these methods are mostly designed in horizontal fashion, meaning that the separate models are coupled at the horizontal interface between the 2D and 1D regions. These methods cannot be adapted for different flooding scenarios and are not able to compute a 2D flow structure within the channel without solving the full 2D models at all times. In this thesis, we propose coupling methods that aim to overcome these limitations. First, we propose the horizontal coupling method (HCM) in the lines of existing methods. The HCM follows the derivation of coupling terms proposed in [Marin and Monnier, 2009] but we arrive at a slightly different coupling term. Then, a discrete coupling term in closed form and the computation of two lateral discharges over channel cross sections, are proposed. Next, we propose a completely new approach to model coupling; we call it the vertical coupling method (VCM). The VCM is based on vertically partitioning the channel flow into two layers. Flows in the lower and upper layers are assumed to be 1D and 2D respectively, and the appropriate flow models derived. By preserving conservation and 1D consistency, the numerical algorithm for coupling the two layer models, is formulated. We show that (i) both the HCM and the VCM are well-balanced and preserve the no-numerical flooding property, (ii) that the VCM adapts to the flow situation: solving the upper layer 2D model only if flooding. Numerical experiments show that both methods provide promising results and that the VCM is able to compute the 2D flow structure within the channel, whenever flooding. We also discuss (without numerical details) how the VCM is a family of methods and some areas of possible further research work are suggested.

Abbreviations

- NSE Navier-Stokes Equations.
- FSEE Free Surface Euler Equations.
- SWE Shallow Water Equations.
- SVM Saint Venant Model.
- HRM Hydrostatic Reconstruction Method.
- FBM Flux-Based Method.
- HCM Horizontal Coupling Method.
- VCM Vertical Coupling Method.
- Full 2D 2D Shallow Water Equations.

List of Symbols

$\vec{\boldsymbol{X}} = (x, y) \in \mathbb{R}^2$.

$z \in \mathbb{R}$.

t: Time variable.

$\eta(\vec{\boldsymbol{X}}, t)$: Free surface elevation in 2D.

$\bar{\eta}(x, t)$: Laterally averaged free surface.

$\partial_t u$: The derivative of u with respect to t.

$u = u(\vec{\boldsymbol{X}}, z, t)$: 3D velocity component along x-direction.

$v = v(\vec{\boldsymbol{X}}, z, t)$: 3D velocity component along y-direction.

$w = w(\vec{\boldsymbol{X}}, z, t)$: 3D velocity component along z-direction.

$\vec{\boldsymbol{U}} := (u, v, w)^T$: The velocity vector in 3D.

$\vec{\boldsymbol{u}} := (u, v)^T$: 3D Velocity vector with only the x and y components.

$\nabla_{3D} = (\partial_x, \partial_y, \partial_z)^T$.

$\nabla = (\partial_x, \partial_y)^T$.

$\Delta_{3D} = (\partial^2_{xx} + \partial^2_{yy} + \partial^2_{zz})$.

g: Acceleration due to gravity.

$H(\vec{\boldsymbol{X}}, t)$: Water depth in 2D.

$\underline{H}(x, y)$: Water depth in 1D models.

$\bar{u}(\vec{\boldsymbol{X}}, t)$: x-component of velocity in 2D.

$\bar{v}(\vec{\boldsymbol{X}}, t)$: y-component of velocity in 2D.

$q_x(\vec{\boldsymbol{X}}, t)$: 2D discharge along x-direction.

$q_y(\vec{\boldsymbol{X}}, t)$: 2D discharge along y-direction.

$z_b(x, y)$: 2D bottom topography.

$Z_b(x)$: 1D bottom topograghy.

$z_b^w(x)$: Channel wall elevation.

$B(x, z)$: Channel lateral width.

$y_l(x, z)$: y-coordinate of channel left wall.

$y_l^w(x) := y_l(x, z_b^w(x))$: y-coordinate of channel left wall at the top.

$y_r(x, y)$: y-coordinate of channel right wall.

$y_r^w(x)$: y-coordinate of channel right wal at the top.

$A(x,t)$: Channel cross sectional area.

$Q(x,t)$: Sectionally-averaged discharge.

$\boldsymbol{W} = (A,Q)^T$: Solution vector in 1D channel model.

$\underline{u}(x,t) = \frac{Q}{A}$: Sectionally averaged veloicty.

$\Pi = (H, q_x, q_y)^T$: Solution vector in 2D SWE.

F_1: Flux function along x-direction.

F_2: Flux function along y-direction.

$F = (F_1, F_2)$: The 2D flux vector function in the 2D SWE.

$\mathcal{H}(A, z_b)$: Function that returns height given the wetted cross sectional area and bottom evelevation.

ϕ: 1D numerical flux function.

$\Phi = (\Phi^A, \Phi^Q)$: Coupling term in HCM.

ϕ^{2D}: 2D numerical flux function.

K_i: 1D grid cell.

K_i^N: Subcell of K_i on the North side.

K_i^S: Subcell of K_i on the South side.

$(\boldsymbol{W})_i^N, (\boldsymbol{W})_i^S$: 2D solution vectors in K_i^N, K_i^S respectively.

$(T_{ij})^N$: 2D cell adjacent to 1D cell K_i on the North side.

$(T_{ij})^S$: 2D cell adjacent to 1D cell K_i on the South side.

$(\Pi)_{ij}^N, (\Pi)_{ij}^S$: 2D solution vectors in $(T_{ij})^N, (T_{ij})^S$ respectively.

$\beta(\vec{\boldsymbol{X}})$: Channel depth.

$A_c(x)$: Area equal to that of an exacly filled channel.

$\eta_1(x,t)$: Top elevation of lower layer in the VCM.

$h_1(\vec{\boldsymbol{X}}, t)$: Lower layer water depth in VCM.

$A_1(x,t)$: Wetted crossectional area of the lower layer.

$Q_1(x,t)$: Sectionally averaged discharge of the lower layer.

$\underline{u}_1 = \frac{Q_1}{A_1}$: Sectionally averaged velocity in the lower layer.

$h_2(\vec{\boldsymbol{X}}, t)$: Water depth in the upper layer.

$\vec{q}_2$: 2D discharge vector in the upper layer.

η_i^β: discrete value of $z_b^w(x)$ in cell K_i.

T_{ij}: Upper layer 2D channel cell corresponding with the lower layer cell, K_i.

$\mathcal{A}$: Computes the total water area in a channel cross section given the free surface distribution.

β_{ij}: Channel dept in upper layer cell T_{ij}.

Δy_{ij}: Width of 2D channel cell, T_{ij}.

S_{ij}: Mass exchange between upper layer cell, T_{ij} and the lower layer 1D cell K_i.

S_i^Q: Total momentum exchange between the lower layer and all the 2D upper layer cells, T_{ij}.

$A_{2,i}$: Sum of all wetted area in the upper layer.

$A_{1,i}$: Wetted cross sectional area in the lower layer.

$A_i = A_{1,i} + A_{2,i}$.

$\vec{u}_{\eta_1,i,j} = (u_{\eta_1,i,j}, v_{\eta_1,i,j})^T$: The interface velocity.

$A_{1,i}^{n+1*}$: Intermediate solution.

Chapter 1

Introduction

1.1 Introduction

All over the world, flooding is a recurring natural disaster that has led to the loss of over 500,000 human lives [Doocy et al., 2013] and caused the damage of properties worth over 80 billion dollars [Smith and Katz, 2013]. In most flooding events, basic facilities such as transportation and communication systems, water and power lines, schools and offices, among others, are usually destroyed while survivors are temporarily or permanently displaced from their homes. [Doocy et al., 2013] reviewed flooding events in the period of 1980 to 2009. Their findings indicate that 539,811 deaths and 361,974 injuries were recorded and 2,821,895,005 people affected in one way or the other due to flooding within the period under review [Doocy et al., 2013]. In 2010, several countries and regions, such as Central Europe, North Eastern Brazil, South West of China, etc, were severely hit by flood disasters [Wang, 2011]. In 2012, Nigeria experienced one of the country's worst flood incidences as several states of the country were devastated by flood [Agbonkhese et al., 2014]. According to the account in [Tawari-Fufeyin et al., 2015], Cameroun experienced sustained rainfall between June and September, 2012 which caused excessive flooding around the Ledja Dam and led to the break of the dam. Hence, water flowed into Nigerian seas through River Benue, through River Niger, causing eleven states (about one-third of the country) to be flooded [Tawari-Fufeyin et al., 2015].

Unfortunately, important human activities, like urbanisation and deforestation, contribute to increased frequency of flooding; in Urbanisation, impermeable materials like tarmac and concrete are used for roads, while deforestation involves the reduction of vegetation cover, see [Jackson, 2016]. Both of these prevent rain water from infiltrating into the soil, so water gets into the rivers more quickly which

may lead to river flooding, see [Jackson, 2016] Climate change is also another factor that increases the risk of flooding [Pitt, 2007]. Therefore, there does not seem that there is an end in sight to flooding. On the other hand, flooding is advantageous with regards to improving soil fertility, natural irrigation, propagation and survival of some species of plants and shrubs, maintenance of the ecosystem, among others [Kharat, 2009].

Consequently, it is environmentally unfriendly and technically not possible to totally eliminate flooding. Therefore, an effective way of managing flood risks are being established [Kharat, 2009]. Flood risk management is established for this purpose. The aim is to design comprehensive plan to reduce the possibility and/or effects of floods, which might include prevention, monitoring, recovery, preparedness and control of flood risk [Wang, 2011].

An important component of flood risk management, as also recommended in [Pitt, 2007], is the development of techniques and tools for predicting floods [Kharat, 2009; Wang, 2011]. The insights and information gained from such predicting tools can be used to develop flood maps, carry out development planing, prepare emergency plans, undertake risk assessment based on susceptibility data for various socio-economic factors, and for learning lessons on sources and courses of flooding [Kharat, 2009]. The information can also facilitate communication of flood risk among stakeholders such as professionals, politicians, public and other interest groups [Pender and Néelz, 2007]. Finally, they can be used for disaster education and evacuation planning and training.

In order to develop such predicting tools, physical (experimental) models or mathematical (computer) models can be used [Kharat, 2009; Wang, 2011]. However, flood-type experiments could be costly, time consuming and difficult to be reused in different scenarios. Hence, mathematical-computer models are viable option [Kharat, 2009]. This thesis, is therefore concerned with developing computationally efficient computer-based models for simultaneously simulating the flow and flooding of rivers. These methods are what is referred to as coupling methods throughout this thesis and in the literature also.

The rest of this chapter is organised as follows. In section 1.2, we briefly describe the basis for the derivation of the mathematical equations usually adopted for flood simulations. We leave the details of the derivations for chapter 2. In section 1.3, we briefly explain the main idea in coupling methods. This is to make the graphics and discussions in the subsequent chapters easy to understand. We state the specific objectives of the research in section 1.4 and review coupling methods in section 1.5. An outline of the thesis is given in section 1.6.

1.2 Flood Flow Simulation

Having identified the need for flood simulations using computer-based methods, we now briefly discuss the approach with which the mathematical equations governing flood flows are derived. We leave the details for a later chapter. As the region occupied by flood can be large compared to the height of flood, we can consider flood flows as shallow water flows. This means that the vertical height of the fluid is very small compared to the horizontal length, so the vertical accelerations are neglected and fluid pressure is taken to be hydrostatic [Stoker, 1957; Toro, 2001; Aldrighetti, 2007; Decoene et al., 2009]. This leads to the shallow water equations which have been widely accepted for modelling shallow flows such as flood flows, river flows, tsunamis, etc [Toro, 2001].

Several researchers have used the shallow water equations for flood simulations, see [Mignot et al., 2006; Hunter et al., 2008; Liang, 2010; Wang, 2011] for examples. However, we are particularly interested in river flooding, that is floods caused by river overflow. River flooding occurs when water enters into the river causing its discharge to exceed its capacity, hence the river overflows onto the surrounding environments [Jackson, 2016]. These surrounding environments which can be flooded by water from the river, are called the floodplains. River flooding is very common and rarely absent in any flooding event since they are triggered once too much water enters into the river. A typical example is the Nigerian incidence of 2012 [Agbonkhese et al., 2014; Tawari-Fufeyin et al., 2015].

To simulate river flooding, it is necessary to simulate both the flow of water along the river channel and flow of water in the floodplains. Assuming the task is to simulate just the river flow alone, then one could assume that river flow is dominated along the river course neglecting variations in flow velocities and free-surface along the vertical and lateral directions, leading to the one-dimensional (1D) St. Venant's model [Stoker, 1957; Cunge et al., 1980; Decoene et al., 2009]. This 1D model is computational inexpensive [Bladé et al., 2012; Morales-Hernández et al., 2013]. However, during flooding when the river overflows, one needs to simulate the flows in both the river and the floodplains. In this case, the 1D model is no longer valid as the flow, especially in the floodplains, has become multidimensional. The problem then arises that even a two-dimensional (2D) simulation of flows in both floodplains and channel, is computationally expensive due to more complicated equations than 1D, data requirements and smaller time steps [Bladé et al., 2012; Morales-Hernández et al., 2013].

Therefore, we run into a deadlock, namely 1D simulation is inaccurate for the

whole domain while 2D simulation is computationally expensive. Fortunately, the 1D assumptions could be retained along the river channel [Fernandez-Nieto et al., 2010]; moreover, the 2D floodplains might be small compared to the entire domain. Therefore, an approach is to decouple the domain, use the 1D model in the river channel and use the 2D model in the floodplains [Fernandez-Nieto et al., 2010; Bladé et al., 2012; Morales-Hernández et al., 2013; Goutal et al., 2014]. The problem of efficiently coupling the two sub-models then arises. This has been the subject of much research, as we review in section 1.5 below, and also the goal of the current research. Therefore, the objective of this thesis is to develop methods to efficiently couple a 1D river model with a 2D flood model for simulating floods.

1.3 The Main Ideas in Coupling Channel and Floodplain Flow Models

In this section, we present some graphics to briefly explain the main idea in coupling channel and floodplain flow models. Imagine the flow of water over a fixed 2D horizontal domain comprising of a channel and floodplains, see figure 1.1. Figure

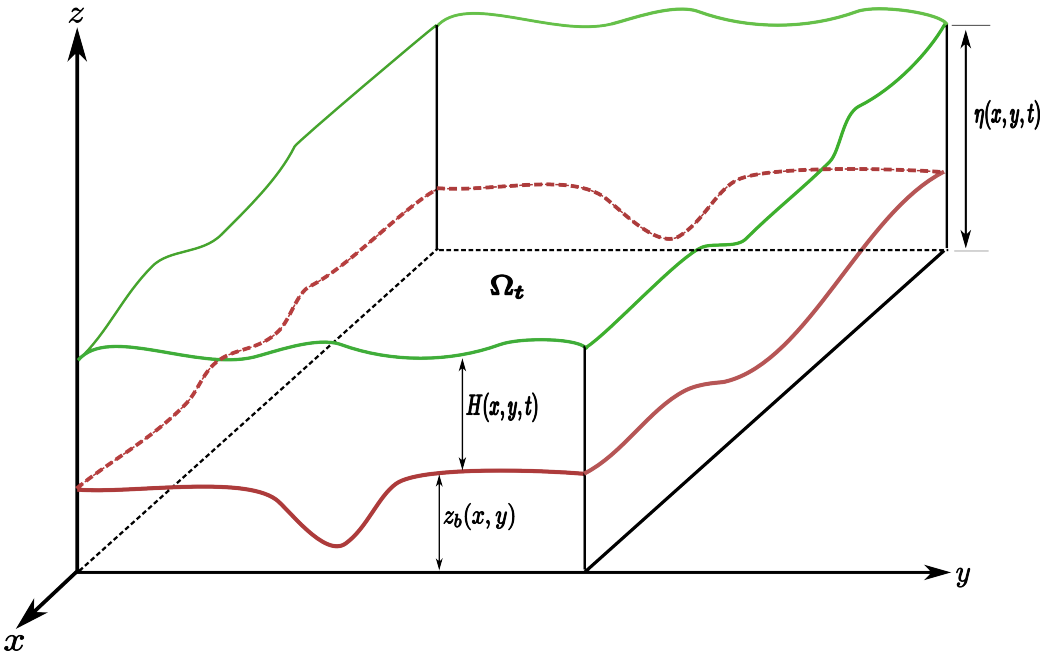


Figure 1.1: Shallow water flow over a domain with bottom topography $z_b(x, y)$ (in red) comprising of a channel and floodplain. The length of the channel is along the x-axis and the width, along the y-axis. The water depth is $H(x, y, t)$ and the free surface elevation is $\eta(x, y, t)$ (in green). Symbol t is time variable and $(x, y, z) \in \mathbb{R}^3$.

1.2 shows a cross section of the flow domain in figure 1.1 at a fixed x. For this cross section the channel is in the region $y_a \leq y \leq y_b$ while the floodplain occupies the remaining region.

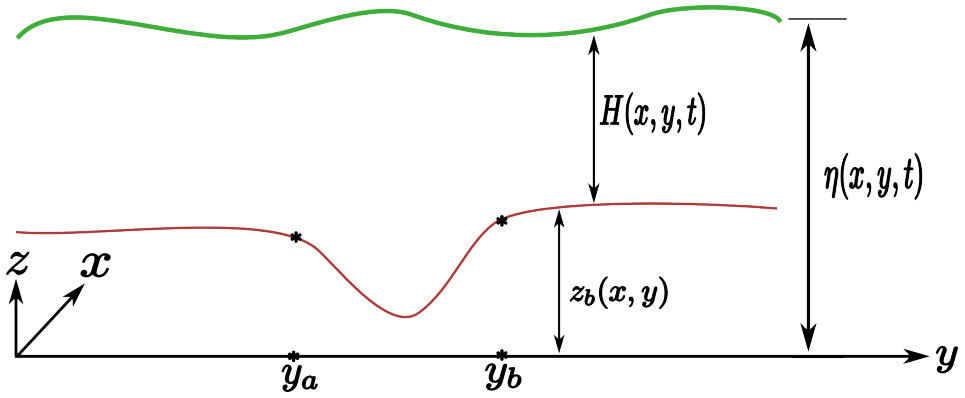


Figure 1.2: Cross section at a fixed point x in the flow domain depicted in figure 1.1. The bottom topography (in red) comprises of the channel in the region $y_a \leq y \leq y_b$ and the floodplain which occupies the remaining regions. The free surface elevation is in green and $H(x, y, t)$ is the water depth.

The idea of coupling methods is to decouple the cross section in figure 1.2, into the channel region and the floodplain regions. This is illustrated in figure 1.3.

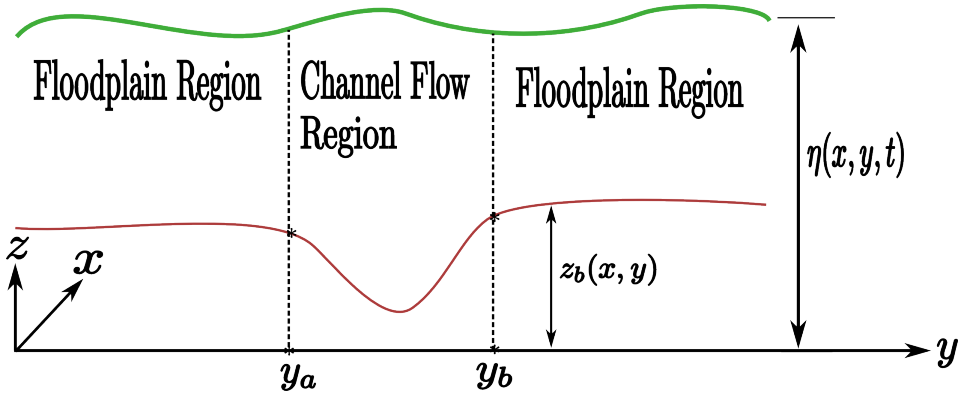


Figure 1.3: Cross section at a fixed x in the flow domain showing the different regions for channel and the floodplains. The middle region is the channel which occupies the region $y_a \leq y \leq y_b$ while the left and right regions are the floodplains.

Then, the 2D shallow water flow model is applied to the flow in the floodplains, while a channel flow model is used for the flow in the channel region. Figure 1.4 illustrates this idea. Since the standard 1D channel flow model (presented in chapter 2.4) does not account for the interaction between the flows in the channel and floodplains nor recover the 2D flow structure within the channel, the main goal

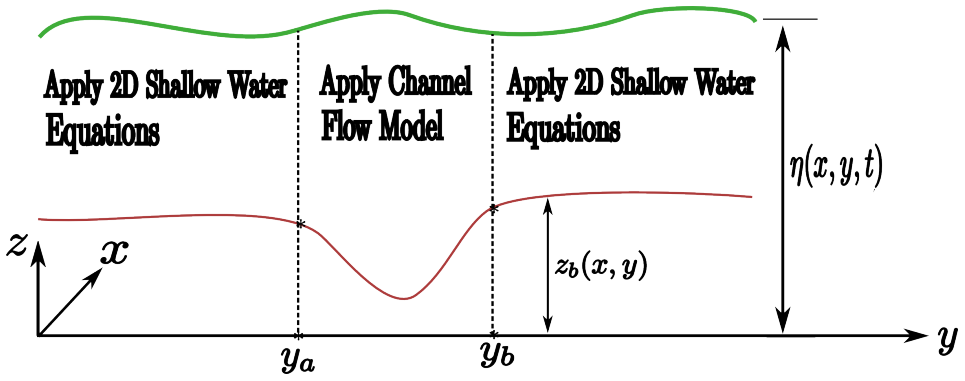


Figure 1.4: A picture of the different models for the different regions. The 2D shallow water equations for the floodplain flow (left and right) and a channel flow model for the channel region (middle).

of coupling methods is to propose models for the channel flow. Different methods propose different models for the channel flow. Therefore, the channel flow model is what actually defers for different coupling methods. Hence, it is usually necessary to depict the problem by showing only the channel cross section without the need to include the floodplains, see figure 1.5. Therefore, in all the methods we propose in this thesis, we depict only the channel flow region. For instance, the horizontal coupling method (HCM) proposes the standard Saint Venant open channel model including a coupling term and the use of the y-component discharge equation for the channel flow. This is depicted in 1.6(a). While the vertical coupling method proposes two-layer models for the channel flow, see figure 1.6(b). Details of these methods are given in subsequent chapters.

The specific objectives of this thesis are listed in the following section.

1.4 Research Objectives

This thesis aims to propose and implement methods to coupled the 2D shallow water equations with the 1D St Venant open channel model that will allow for the efficient and accurate simulation of river flooding. It is also intended to validate the methods using some hypothetical test cases. Specifically, this thesis aims to achieve the following objectives.

- Implement and test a 2D numerical solver for the 2D shallow water flow models using a well-balanced method.
- Implement and test a 1D numerical solver for the 1D St Venant open channel model.

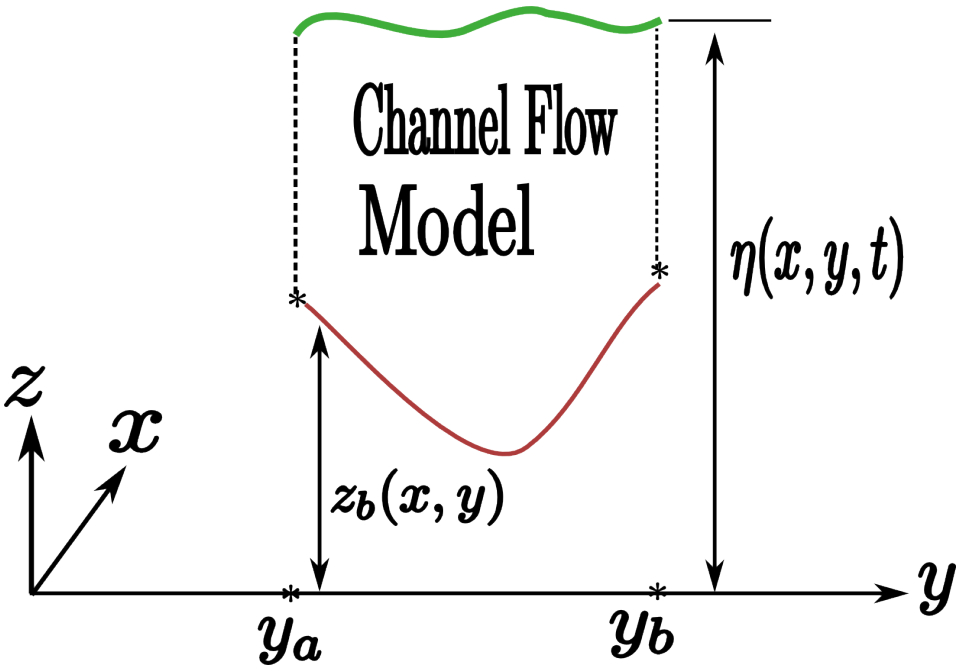
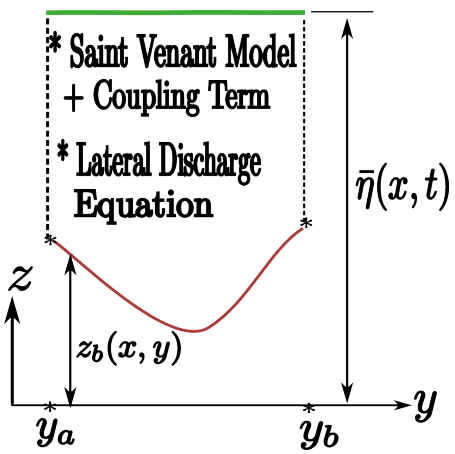
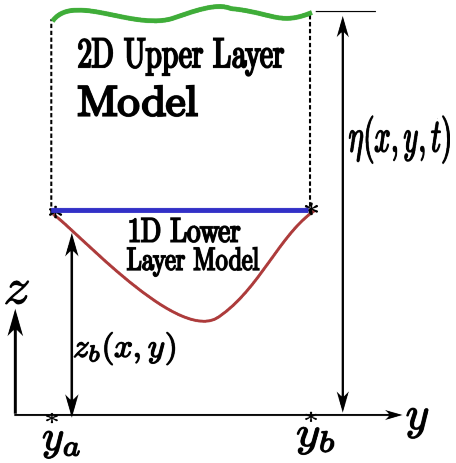


Figure 1.5: Flow cross section depicting only the channel region (channel cross section). Channel model is derived for the flow in this region. The flow in the floodplains are used as boundary conditions. The channel flow also serve as boundary conditions for the 2D floodplain models.



(a) Physical structure of the channel flow in the horizontal coupling method. The channel flow is modelled with (i) The Saint Venant Model with coupling term and (ii) the lateral discharge equation in the 2D shallow water equations



(b) Physical structure of the channel flow in the vertical coupling method. The channel flow is described with two-layer models - the upper layer model and lower layer model. The blue line separates the two layers.

Figure 1.6: Channel flow structure for the horizontal coupling method (left) and the vertical coupling method (right).

- Implement the 2D/1D coupling method proposed in Bladé et al. [2012], compare the result with full 2D solver, then propose and implement a coupling method following the lines of the mathematical formulations in [Marin and Monnier, 2009].
- Propose and implement a completely new coupling method based on vertical partitioning of channel flow.
- Validate all the proposed methods using hypothetical data.

In the following section, a review of some existing coupling methods is presented.

1.5 Review of Coupling Methods

As mentioned above, coupling methods aim to utilise the advantages of the computational efficiency of the 1D river model and the accuracy of the 2D flood model. Early attempts to couple river and floodplain flows were developed as extension of existing 1D river models where the floodplain is modelled by using storage areas, see [Cunge et al., 1980; Bladé et al., 1994] for example. But this approach does not allow to study the dynamics of flow in the floodplains [Fernandez-Nieto et al., 2010]. Hence, methods which allow to simultaneously simulate the flow dynamics in both the river and floodplains are being developed.

In [Bladé et al., 2012], a method for numerically coupling the 1D model and 2D was developed. The authors used the 1D St Venant model for the main river channel and the 2D shallow water model for floodplain flows. Coupling is then achieved through numerical fluxes in the finite volume discretization of the models. They called the method, the Flux-Based Method (FBM).

Another idea which uses the theory of characteristics to formulate a 1D and 2D coupling method is presented in [Chen et al., 2012]. In this approach, the 1D and 2D models are the same as those in [Bladé et al., 2012]. The 1D model was solved with the implicit four-point Preissmann scheme while the 2D model was solved with a PISO-like algorithm. They considered only boundary-to-boundary (frontal) coupling. To couple the two models, they defined some matching conditions at the boundary points. Then a prediction and correction algorithm was designed to ensure that the matching conditions are satisfied. The 1D and 2D shallow water model each coupled with the sediment transport model was studied in [Zhang et al., 2014]. The authors used explicit finite volume method with the HLL Riemann solver for all models.

In another study [Seyoun et al., 2012], the 1D river model was coupled with the 2D non-inertia model to simulate the interaction of a sewer system with overland flow. The authors used the 1D model implemented in SWMM5 [Rossman et al., 2005] which is a finite difference approximation of the 1D shallow water system. They solved the 2D model using the alternating direction implicit(ADI) finite difference scheme. The discharges at the 2D/1D interface are calculated according to the water level differences between the flows in the two domains. This study used the non-inertia 2D models which ignore the convective terms in the shallow water models.

Two different methods of coupling 1D and 2D models were developed in [Morales-Hernández et al., 2013], see also [Morales-Hernández, 2014]. The methods are also based on existing 1D and 2D models. The two models are solved with explicit finite volume methods. The authors defined what is called the coupling zone which is an intersection line between the sub-domains. They called the methods the only mass conservation(OMC) and mass momentum conservation (MMC) methods. Both methods are based on post-processing of separately computed solutions of the existing 1D and 2D models. For both methods, each model(1D and 2D) computes its own time step and the minimum is chosen as the current time step. For the OMC method, each model updates its own cell averages independently. The updated values in the 2D and 1D cells at the coupling zone do not account for the interaction between the two sub-domains. Hence, these updated values are regarded as **star solutions** at the coupling zone. With these **star solutions**, the total water volume in a 1D cell and all its adjacent 2D cells is computed from which a common water level is found for all the cells. Using this common water level, the water height for 2D cells and the wetted area for the associated 1D cell are found. These are taken as the new cell averages for the new time level. For the MMC approach, in addition to the above procedure, a similar approach is also adopted to enforce momentum conservation which then enables to compute the new discharges for all the 1D and 2D cells. The possible issues with these methods are: (i) The post-processing step, which ensures mass and/or momentum conservation, might not guarantee the accuracy of the solution since it computes the free-surface to be laterally constant in each 1D cell and its adjacent 2D cells and; (ii) The first step where the conserved variables were updated to get star values, may give inadequate results at coupling zone since the calculation completely ignores the interaction between the two sub-domains. Recently, these strategies were applied to Tiber River, Rome in [Morales-Hernández et al., 2016].

The superposition method is proposed in [Marin and Monnier, 2009]. The

authors classically derived the exchange terms in the 1D model from the full 3D Navier-Stokes equations and numerically coupled the 1D and 2D models using an optimal control process. In [Fernandez-Nieto et al., 2010], the superposition approach of [Marin and Monnier, 2009] was extended to finite volume methods in which a discrete exchange term that leads to globally well-balanced scheme, was derived. In this approach, a 2D grid is superposed over the 1D channel grid and a Schwartz-like iterative algorithm is used to achieve convergence. The challenge in this method is that in practical cases, the iterative process can jeopardise the overall efficiency of the method [Goutal et al., 2014].

In all the coupling methods, a great difficulty is the calculation of the lateral discharge along the river channel since the 1D model does not have an equation to compute it. In [Ghostine et al., 2015], the lateral discharge was set to zero and used in evaluating 2D numerical fluxes at the interfaces between 1D region and 2D region. Using the models and exchange terms derived in [Marin and Monnier, 2009], a strategy to estimate this lateral discharge without superposition or overlapping, was proposed in [Goutal et al., 2014]. The approach uses an iterative technique to estimate the transverse velocity. This iterative technique uses the solution of successive Riemann problem. However, the problem of computing the lateral or transverse discharge/velocity remains a major challenge.

Different classes of coupling approaches have been applied to solve coupled continuum mechanics problems. Among them are the closely or fully coupled methods which at each time step, solve the governing equations simultaneously for all the sub-domains, while iterative coupled methods solve the equations iteratively until convergence is achieved [Von Estorff and Hagen, 2006; Settari and Walters, 2001]. Most of the methods for river-flood coupling are of the closely coupled type, like those in [Bladé et al., 2012; Morales-Hernández, 2014; Ghostine et al., 2015; Goutal et al., 2014]. A few examples also exist for iterative type methods, like the superposition method [Marin and Monnier, 2009; Fernandez-Nieto et al., 2010]. In practical cases of river-flood coupling, iterative process might not converge and can jeopardise the overall efficiency of the method [Goutal et al., 2014]. Therefore, all the methods considered in this thesis are of the closely coupled type.

In the context of shallow water equations, [Morales-Hernández et al., 2013] classified coupling methods into two types, namely lateral and frontal coupling. While lateral coupling refers to methods that couple 1D and 2D models in subdomains that are connected at the lateral boundaries of the channel; frontal coupling refers to methods that couple the models when the connecting interface is in direction of the channel flow. The lateral and frontal coupling methods both face the difficulty

in computing the lateral discharge mentioned above.

Lateral coupling methods retain 1D assumptions in the channel- that velocity and free surface are laterally uniform - even during flooding. Hence, these methods are not able to compute the true 2D flow structure in the channel during flooding. On the other hand, frontal coupling methods are able to compute 2D flow structure in the channel but they loose efficiency because they compute the 2D solutions at all times even when flooding seizes to occur. Hence, we need methods that can compute 2D flow structure in the channel only during flooding and revert back to compute 1D solution when flooding seizes.

In addition to the above limitations, most of the existing methods require the knowledge of flooding regions a priori. In reality, flooding locations would vary with time, so might be impossible to know a priori. Another issue is that majority of existing methods are either lateral or frontal type, not being able to automatically switch types. Finally, methods that are superset of several existing methods are desirable. Hence, it would be important to develop methods that :

1. Compute the 2D flow structure within the channel during flooding but revert back to 1D simulation if not flooding.

2. Do not require a priori knowledge of the flooding locations but are able to automatically detect them.

3. Automatically resolve both frontal and lateral type flooding problems without having to use multiple methods.

4. Are superset of existing methods, reducing to different methods under different choices of model parameters.

In this thesis, two coupling methods are proposed not only for the purpose of achieving good accuracy and efficiency but also to achieve the goals listed above. The first one is based on the derivation in [Marin and Monnier, 2009]. In this method, we derive a model similar to the one in [Marin and Monnier, 2009] and propose to compute the transverse discharge using the third equation of the 2D shallow water equation. The second method, called the Vertical Coupling Method (VCM), is a completely new approach where the transverse velocity is automatically computed, flooding regions are automatically detected and it is a superset of different methods. Well-balanced schemes are schemes that satisfy steady state solutions at discrete level [Greenberg and Leroux, 1996]. A coupling method is said to satisfy the no-numerical flooding property if it does not erroneously produce flooding when the channel is not full (introduced in section 4.6). These two properties are important

for the stability of a coupling method. The methods proposed in this thesis are shown to be well-balanced and satisfy the no-numerical flooding property.

The scope of this project is to propose these methods, prove their properties, numerically show their accuracy compared with the FBM, and show that the VCM computes 2D solutions in the channel only during flooding. Then, we briefly explain (without numerical details) how the VCM is a unifying method. The thesis does not consider lateral coupling methods. These and other issues are listed as suggestions for further work.

1.6 Outline of the Thesis

The outline of the thesis is the following. In chapter 2, the 2D shallow water flow model and the 1D St Venant open channel flow model are separately derived for the case where there is no linking between the two models. The procedure adopted is to start from the three-dimensional (3D) Navier Stokes equations (NSE), then obtain the 3D free-surface Euler's equations (FSEE) and finally obtain the 2D SWE. Then the FSEE is revisited from which the 1D St Venant model (SVM) for open channel is obtained. A few mathematical properties of these equations are briefly discussed.

In chapter 3, an overview of the numerical methods for the models derived in chapter 2 is presented. Starting from discussing conservative numerical methods, we introduce the finite volume methods for 1D system of conservation laws. Then we outline one existing finite volume method for the channel flow model and one for the 2D SWE (the flood models). We present a few numerical experiments to verify our implementation of the methods discussed in the chapter. We also show numerically that the 1D assumption along the channel when not overflowing is valid, justifying the need for model coupling. Furthermore, we show that the 2D flow structure within the channel during flooding is not completely absent, motivating the vertical coupling method.

Chapter 4 presents our first coupling method, the HCM. We motivate the approach by explaining why it is important not to always assume that the channel lateral discharges are zero or even constant during flooding. Following the approach in [Marin and Monnier, 2009], we revisit the derivation of the SVM in chapter 2 and now assume that the channel is full but still maintaining a laterally flat free surface. This allowed us to derive St Venant model with coupling term. Then we formulate a finite volume scheme for the model including the discrete coupling term in closed form. We also propose to solve the y-discharge equation in the 2D SWE for the channel lateral discharges. We then introduce the concept of no-numerical

flooding and prove that the proposed method satisfies this property and is also well-balanced. We use the term no-numerical flooding property to mean that a coupling method does not produce artificial (numerical) flood when the channel is not full. One numerical experiment is then used to access the performance of the method. More numerical experiments are carried out in chapter 6. All numerical experiments show that the method gives very promising results.

In chapter 5, we propose our new coupling method, the VCM. Unlike the derivation of the HCM, here we take the channel to be full and that the free surface is **not** laterally flat. This means that we allow a true 2D flow structure. But we require that the channel free surface be laterally flat when the channel is not full. This allows us to return to 1D simulations should the channel not be full. The channel is then partitioned vertically into two layers. The lower layer flow is assumed to be 1D and an appropriate 1D model with exchange term is derived. Similarly, the upper layer flow is assumed to be 2D and the appropriate 2D model with exchange term is derived. The numerical implementation is then detailed. The numerical implementation consists of the following three steps. The first step is to distribute a given 2D solution among the two layers within the channel, the second step is to evolve each layer data using their respective models and compute the lateral flow exchange between the channel and the floodplains. The third and last step combines the evolved data from the separate layers to evolve the flow to new time level. The details of these are all presented. The method is proved to be well-balanced and preserve the no-numerical flooding property.

Chapter 6 presents numerical experiments to access the performance of the methods proposed in this thesis. We remark that our goal is not for the coupling methods to reproduce experimental results but to accurately approximate the results of a full 2D simulation. The full 2D model has been widely used to model flooding, see for example [Mignot et al., 2006; Hunter et al., 2008; Liang, 2010; Wang, 2011], hence we use it as the reference solution for all the cases considered in this thesis. The results show that the proposed methods are indeed adequate when compared with the full 2D results. The results also show that the VCM recovers the 2D flow structure within the channel during flooding. The thesis is concluded in chapter 7 where conclusions are made and possible areas of further research work suggested.

Chapter 2

Mathematical Models for Flood and Channel Flows

2.1 Introduction

The general mathematical model governing the flow of a fluid in many applications, in which the continuum hypothesis is valid, is the Navier-Stokes equations (NSE). Such applications include flows in pipes, seas, oceans, around aircraft and flooding. For a given application, the NSE are complimented with boundary and initial conditions, which in principle, should enable to completely solve the equations for the problem under investigation. However, for free-surface flows such as flood and channel flows, amongst many others, the position of the boundary of the flow domain is usually not known a priori. This makes a direct application of the NSE to free-surface flow problems difficult as the boundary conditions can not be directly specified. In addition to the above difficulty, solving the full 3D NSE is computationally expensive as there does not exist an exact analytical solution for the general 3D NSE. Therefore, to solve a free-surface flow problem, further model derivation is usually carried out, with the hope of simplifying the NSE into less computationally expensive models and also to circumvent the unknown boundary position problem.

For some flows, like those in seas, rivers, floods, atmosphere, and open channels, amongst many others, the horizontal length scales (like river length) are much larger than the vertical length scales, like fluid depth [Stoker, 1957; de Boer, 2003]. This allows to assume that vertical component of acceleration is negligible, which in turn, leads to the assumption of hydrostatic pressure distribution, which means that the net pressure exerted on a fluid particle is only due to the force exerted on it by other fluid particles lying above it (see for example [Toro, 2001; Cunge et al., 1980]).

With the above approximation, the shallow water theory is established where the unknown free-surface position, $\eta(\vec{\boldsymbol{X}}, t)$ is formulated as part of the solution of the problem. The aim of this chapter is to derive the 2D Shallow Water Equations and the 1D Saint Venant Model needed to simulate flood and open channel flows respectively, in the following chapters. The chapter also provides the foundation for all the derivations in the following chapters.

The rest of this chapter is organised as follows. Starting with the NSE, the 3D free-surface Eulers Equations (FSEE) is derived in section 2.2, then the 2D shallow water equations (SWE) is derived in section 2.3. Section 2.4 presents the derivation of the 1D Saint Venant Model (SVM) for open channel with arbitrary shaped geometry, from which the special SVM for a locally rectangular cross-sectional channel, and also the constant width 1D SWE are obtained. Then the chapter is concluded in section 2.5 where it is briefly shown that the models derived in the previous sections are hyperbolic conservation laws with source terms, also referred to as balance laws. Some important properties of such partial differential equations that necessitate their types of methods of solution, are also briefly outlined.

2.2 The Free-Surface Euler Equations

While the NSE is the model for general fluid problems, the free-surface Euler equations (FSEE) is the model which govern the flow of an incompressible, inviscid flow under gravity. In this section, we derive the FSEE from the incompressible NSE.

2.2.1 Background

Consider the flow of an incompressible fluid which at time, t occupies the domain, Ω_t defined by

$$\Omega_t = \{(\vec{\boldsymbol{X}}, z) \in \mathbb{R}^3 : \vec{\boldsymbol{X}} = (x, y) \in \Omega_H \subset \mathbb{R}^2 \text{ fixed}, z_b(\vec{\boldsymbol{X}}) \leq z \leq \eta(\vec{\boldsymbol{X}}, t)\}. \quad (2.1)$$

The domain is bounded below by a fixed bottom, $z_b(\vec{\boldsymbol{X}})$ and above by the free-surface position, $\eta(\vec{\boldsymbol{X}}, t)$ given by

$$\eta(\vec{\boldsymbol{X}}, t) = z_b(\vec{\boldsymbol{X}}) + H(\vec{\boldsymbol{X}}, t). \quad (2.2)$$

where $H(\vec{\boldsymbol{X}}, t)$ is the depth of fluid at time, t, see figure 2.1.

The flow of an incompressible Newtonian fluid in Ω_t is governed by the

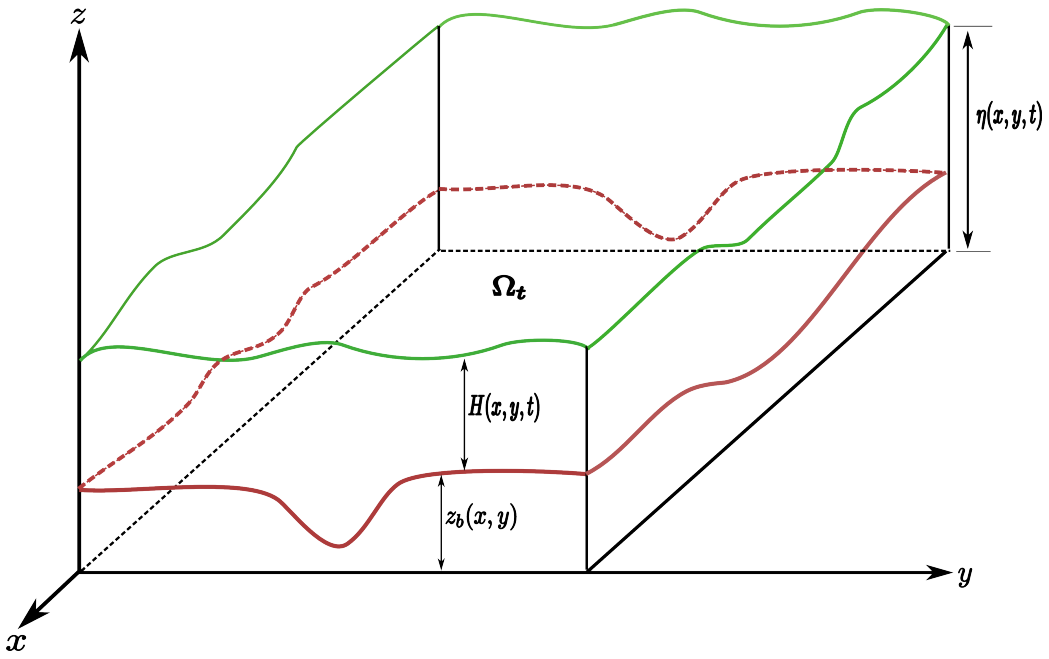


Figure 2.1: Shallow water flow over a domain with bottom topography $z_b(x, y)$ (in red). The water depth is $H(x, y, t)$ and the free surface elevation is $\eta(x, y, t)$ (in green). Symbol t is time variable and $(x, y, z) \in \mathbb{R}^3$.

Navier-Stokes Equations (NSE), namely

$$\nabla_{3D} \cdot \vec{\boldsymbol{U}} = 0. \tag{2.3}$$

$$\partial_t \vec{\boldsymbol{U}} + \left(\vec{\boldsymbol{U}} \cdot \nabla_{3D}\right) \vec{\boldsymbol{U}} = -\frac{1}{\rho} \nabla_{3D} P + \nu \Delta_{3D} \vec{\boldsymbol{U}} + \vec{\boldsymbol{g}}. \tag{2.4}$$

where $\vec{\boldsymbol{U}} = \vec{U}(\vec{\boldsymbol{X}}, z, t) = \left(u(\vec{\boldsymbol{X}}, z, t), v(\vec{\boldsymbol{X}}, z, t), w(\vec{\boldsymbol{X}}, z, t)\right)^T$ and $P = P(\vec{\boldsymbol{X}}, z, t)$ are the fluid velocity and pressure respectively at the point $(\vec{\boldsymbol{X}}, z) \in \Omega_t$ at time, t. While ν and ρ are the fluid viscosity coefficient and fluid density respectively and $\vec{\boldsymbol{g}} = (0, 0, -g)^T$, where g is the constant acceleration due to gravity. $\nabla_{3D} = (\partial_x, \partial_y, \partial_z)^T$ and $\Delta_{3D} = (\partial^2_{xx} + \partial^2_{yy} + \partial^2_{zz})$.

2.2.2 Deriving the free-surface Euler Equations

To derive the FSEE, the following assumptions are made [Lannes, 2013].

i The fluid is homogeneous and inviscid.

ii The fluid particles do not cross the bottom and the free-surface.

iii There is no surface tension and the external pressure is constant.

The first assumption means that the NSE, (2.3)-(2.4) reduce to the inviscid Euler Equations (2.5)-(2.6) below, while the second assumption implies that the kinematic boundary condition holds on both the bottom and free-surface, namely equations (2.7) and (2.8) respectively. And the last assumption gives rise to the dynamic boundary condition on the free-surface, namely equation (2.9).

$$\nabla_{3D} \cdot \vec{\boldsymbol{U}} = 0. \tag{2.5}$$

$$\partial_t \vec{\boldsymbol{U}} + \left(\vec{\boldsymbol{U}} \cdot \nabla_{3D}\right) \vec{\boldsymbol{U}} = -\frac{1}{\rho} \nabla_{3D} P + \vec{\boldsymbol{g}}. \tag{2.6}$$

$$\Big((\vec{\boldsymbol{u}} \cdot \nabla) z_b(\vec{\boldsymbol{X}}) - w(\vec{\boldsymbol{X}}, z, t)\Big)\Big|_{z=z_b(\vec{\boldsymbol{X}})} = 0. \tag{2.7}$$

$$\Big(\partial_t \eta(\vec{\boldsymbol{X}}, t) + (\vec{\boldsymbol{u}} \cdot \nabla)\eta(\vec{\boldsymbol{X}}, t) - w(\vec{\boldsymbol{X}}, z, t)\Big)\Big|_{z=\eta(\vec{\boldsymbol{X}},t)} = 0. \tag{2.8}$$

$$P(\vec{\boldsymbol{X}}, z, t) = P_{atm} \quad \text{on } z = \eta(\vec{\boldsymbol{X}}, t). \tag{2.9}$$

Here, $\vec{\boldsymbol{u}} = (u(\vec{\boldsymbol{X}}, z, t), v(\vec{\boldsymbol{X}}, z, t))^T, \nabla = (\partial_x, \partial_y)^T$ and P_{atm} is the atmospheric pressure, which is usually conveniently taken to be zero. We take it to be zero throughout this thesis. In the remainder of this thesis, we shall drop the notations for dependent variables on the 3D velocity components. This is for convenience and to allow for easy read.

Equations (2.5) - (2.9) are known as the Free-Surface Euler Equations (FSEE) [Lannes, 2013]. These are the fundamental equations from which all the models needed in this thesis shall be derived. To derive the models in the subsequent chapters, the continuity and momentum equations (2.5)-(2.6) shall be integrated component by component. For this reason, we also write the FSEE, (2.5) - (2.9), in component form, namely

$$\nabla_{3D} \cdot \vec{\boldsymbol{U}} = 0. \tag{2.10}$$

$$\partial_t u + \vec{\boldsymbol{U}} \cdot (\nabla_{3D} u) = -\frac{1}{\rho} \partial_x P. \tag{2.11}$$

$$\partial_t v + \vec{\boldsymbol{U}} \cdot (\nabla_{3D} v) = -\frac{1}{\rho} \partial_y P. \tag{2.12}$$

$$\partial_t w + \vec{\boldsymbol{U}} \cdot (\nabla_{3D} w) = -\frac{1}{\rho} \partial_z P - g. \tag{2.13}$$

$$\Big(\vec{\boldsymbol{u}} \cdot (\nabla z_b(\vec{\boldsymbol{X}})) - w\Big)\Big|_{z=z_b(\vec{\boldsymbol{X}})} = 0. \tag{2.14}$$

$$\Big(\partial_t \eta(\vec{\boldsymbol{X}}, t) + \vec{\boldsymbol{u}} \cdot (\nabla \eta(\vec{\boldsymbol{X}}, t)) - w\Big)\Big|_{z=\eta(\vec{\boldsymbol{X}},t)} = 0. \tag{2.15}$$

$$P(\vec{\boldsymbol{X}}, z, t) = P_{atm} \quad \text{on } z = \eta(\vec{\boldsymbol{X}}, t). \tag{2.16}$$

2.3 Derivation of 2D Shallow Water Equations

As stated in the introductory section of this chapter, the shallow water theory, commonly referred to as the theory of long waves [Stoker, 1957], assumes that the depth of water is significantly small compared to horizontal length scales such as wave length [Toro, 2001; Lannes, 2013; Stoker, 1957; Tanguy, 2010]. In this section, starting from the FSEE (2.10)-(2.16), we shall derive the Shallow Water equations (SWE), along the lines followed in [Toro, 2001; Tanguy, 2010]. The assumption of small depth in comparison with horizontal length scales, allows to assume that the vertical component of acceleration is negligible, that is

$$\frac{d}{dt} w(\vec{\boldsymbol{X}}, z, t) := \partial_t w + \vec{\boldsymbol{U}} \cdot (\nabla_{3D} w) = 0. \tag{2.17}$$

where we have dropped the independent variable notations for convenience. This means that the left hand side of equation (2.13) in FSEE vanishes, reducing it to the following equation:

$$-\frac{1}{\rho} \partial_z P(\vec{\boldsymbol{X}}, z, t) = g. \tag{2.18}$$

Integrating equation (2.18) with respect to z and using the dynamic boundary condition (2.16), the pressure is obtained as

$$P(\vec{\boldsymbol{X}}, z, t) = \rho g(\eta(\vec{\boldsymbol{X}}, t) - z). \tag{2.19}$$

Equation (2.19) defines the pressure as being hydrostatic which means that the pressure depends only on gravity. Differentiating equation (2.19) with respect to x and y gives

$$\partial_x P(\vec{\boldsymbol{X}}, z, t) = \rho g \partial_x \eta(\vec{\boldsymbol{X}}, t), \quad \partial_y P(\vec{\boldsymbol{X}}, z, t) = \rho g \partial_y \eta(\vec{\boldsymbol{X}}, t). \tag{2.20}$$

The FSEE (2.10)-(2.16) then reduce to the following:

$$\nabla_{3D} \cdot \vec{\boldsymbol{U}} = 0. \tag{2.21}$$

$$\partial_t u + \nabla_{3D} \cdot (u \vec{\boldsymbol{U}}) = -g \partial_x \eta(\vec{\boldsymbol{X}}, t). \tag{2.22}$$

$$\partial_t v + \nabla_{3D} \cdot (v \vec{\boldsymbol{U}}) = -g \partial_y \eta(\vec{\boldsymbol{X}}, t). \tag{2.23}$$

together with the kinematic boundary conditions (2.14) and (2.15). We have used the the vector identity,

$$\nabla_{3D} \cdot (u\vec{\boldsymbol{U}}) = u(\nabla_{3D} \cdot \vec{\boldsymbol{U}}) + \vec{\boldsymbol{U}} \cdot (\nabla_{3D} u)$$

and the continuity equation (2.10) to put the left hand sides of (2.22) and (2.23) in conservative forms.

Notice that the right hand sides of (2.22) and (2.23) are independent of z, hence their left hand sides being the total derivatives, du/dt and dv/dt respectively, are also independent of z. This means that u and v are also independent of z [Stoker, 1957; Toro, 2001]. From this we can conclude that u, v are not very different from their vertically averaged values, $\bar{u}(\vec{\boldsymbol{X}}, t)$ and $\bar{v}(\vec{\boldsymbol{X}}, t)$, so we can decompose them into the sum of their averages and some small perturbations, namely

$$u(\vec{\boldsymbol{X}}, z, t) = \bar{u}(\vec{\boldsymbol{X}}, t) + \widetilde{u}(\vec{\boldsymbol{X}}, z, t). \tag{2.24}$$

$$v(\vec{\boldsymbol{X}}, z, t) = \bar{v}(\vec{\boldsymbol{X}}, t) + \widetilde{v}(\vec{\boldsymbol{X}}, z, t). \tag{2.25}$$

where $\tilde{u}(\vec{\boldsymbol{X}}, z, t)$ and $\tilde{v}(\vec{\boldsymbol{X}}, z, t)$ are very small perturbations whose average, sum and higher order terms vanish.

Define the discharges along x and y directions respectively, namely

$$q_x(\vec{\boldsymbol{X}}, t) = \int_{z_b(\vec{\boldsymbol{X}})}^{\eta(\vec{\boldsymbol{X}},t)} u(\vec{\boldsymbol{X}}, z, t) dz, \tag{2.26}$$

$$q_y(\vec{\boldsymbol{X}}, t) = \int_{z_b(\vec{\boldsymbol{X}})}^{\eta(\vec{\boldsymbol{X}},t)} v(\vec{\boldsymbol{X}}, z, t) dz. \tag{2.27}$$

Integrating the continuity equation (2.21), vertically (over $z_b(\vec{\boldsymbol{X}}) \leq z \leq \eta(\vec{\boldsymbol{X}}, t)$), gives

$$\int_{z_b(\vec{\boldsymbol{X}})}^{\eta(\vec{\boldsymbol{X}},t)} (\partial_x u + \partial_y v + \partial_y w)\, dz = 0.$$

Using the Leibniz rule leads to the following equation:

$$0 = \partial_x \int_{z_b(\vec{\boldsymbol{X}})}^{\eta(\vec{\boldsymbol{X}},t)} u\, dz + \partial_y \int_{z_b(\vec{\boldsymbol{X}})}^{\eta(\vec{\boldsymbol{X}},t)} v\, dz + \Big[w - \vec{\boldsymbol{u}} \cdot (\nabla \eta)\Big]_{z=\eta} + \Big[\vec{\boldsymbol{u}} \cdot (\nabla z_b) - w\Big]_{z=z_b}.$$

We use the definitions of the averages in equations (2.26) and (2.27) to simplify the first two terms on the right hand side. Then, use of the kinematic boundary conditions (equations (2.14) and (2.15)) simplifies the last two terms on the right hand side, hence we have the following equation:

$$\partial_x q_x(\vec{\boldsymbol{X}},t) + \partial_y q_y(\vec{\boldsymbol{X}},t) = -\partial_t \eta(\vec{\boldsymbol{X}},t) = -\partial_t H(\vec{\boldsymbol{X}},t). \tag{2.28}$$

Hence, the following equation, for the depth of water, is obtained:

$$\partial_t H(\vec{\boldsymbol{X}},t) + \partial_x q_x(\vec{\boldsymbol{X}},t) + \partial_y q_y(\vec{\boldsymbol{X}},t) = 0. \tag{2.29}$$

Similarly, if we integrate the x-momentum equation, (2.22) vertically over $z_b(\vec{\boldsymbol{X}}) \leq z \leq \eta(\vec{\boldsymbol{X}},t)$, apply the Leibniz rule, use the kinematic boundary conditions, (2.14) and (2.15) and the definitions (2.24)-(2.27), then the following equation is obtained:

$$\begin{aligned}\partial_t q_x(\vec{\boldsymbol{X}},t) + \partial_x\left(\frac{q_x^2(\vec{\boldsymbol{X}},t)}{H(\vec{\boldsymbol{X}},t)}\right) + \partial_y\left(\frac{q_x(\vec{\boldsymbol{X}},t)q_y(\vec{\boldsymbol{X}},t)}{H(\vec{\boldsymbol{X}},t)}\right) &= -gH(\vec{\boldsymbol{X}},t)\partial_x\eta(\vec{\boldsymbol{X}},t) \\ &= -\partial_x\left(\frac{g}{2}H^2(\vec{\boldsymbol{X}},t)\right) - gH(\vec{\boldsymbol{X}},t)\partial_x z_b(\vec{\boldsymbol{X}}).\end{aligned} \tag{2.30}$$

Finally, if we integrate the y-momentum equation, (2.23) vertically over $z_b(\vec{\boldsymbol{X}}) \leq z \leq \eta(\vec{\boldsymbol{X}},t)$, apply the Leibniz rule, use the kinematic boundary conditions, (2.14) and (2.15) and the definitions, (2.24)-(2.27) then the following equation is obtained:

$$\begin{aligned}\partial_t q_y(\vec{\boldsymbol{X}},t) + \partial_x\left(\frac{q_x(\vec{\boldsymbol{X}},t)q_y(\vec{\boldsymbol{X}},t)}{H(\vec{\boldsymbol{X}},t)}\right) + \partial_y\left(\frac{q_y^2(\vec{\boldsymbol{X}},t)}{H(\vec{\boldsymbol{X}},t)}\right) &= -gH(\vec{\boldsymbol{X}},t)\partial_y\eta(\vec{\boldsymbol{X}},t) \\ &= -\partial_y\left(\frac{g}{2}H^2(\vec{\boldsymbol{X}},t)\right) - gH(\vec{\boldsymbol{X}},t)\partial_y z_b(\vec{\boldsymbol{X}}).\end{aligned} \tag{2.31}$$

After some simplifications, the 2D SWE is summarised in vector form as

follows:

$$\partial_t \Pi + \nabla \cdot F(\Pi) = S(\Pi, z_b),$$
$$\text{where } \Pi = \begin{pmatrix} H \\ q_x \\ q_y \end{pmatrix}, \quad S(\Pi, z_b) = \begin{pmatrix} 0 \\ -gH\partial_x z_b(\vec{\boldsymbol{X}}) \\ -gH\partial_y z_b(\vec{\boldsymbol{X}}) \end{pmatrix},$$
$$F(\Pi) = (F_1(\Pi), F_2(\Pi)), \quad F_1(\Pi) = \begin{pmatrix} q_x \\ \frac{q_x^2}{H} + \frac{1}{2}gH^2 \\ \frac{q_x q_y}{H} \end{pmatrix}, F_2(\Pi) = \begin{pmatrix} q_y \\ \frac{q_x q_y}{H} \\ \frac{q_y^2}{H} + \frac{1}{2}gH^2 \end{pmatrix}. \tag{2.32}$$

The vertically averaged velocity components along the x and y directions are $\bar{u}(\vec{\boldsymbol{X}}, t) = \frac{q_x(\vec{\boldsymbol{X}},t)}{H(\vec{\boldsymbol{X}},t)}$ and $\bar{v}(\vec{\boldsymbol{X}}, t) = \frac{q_y(\vec{\boldsymbol{X}},t)}{H(\vec{\boldsymbol{X}},t)}$ respectively.

2.4 1D Saint Venant Equations for an Open Channel With Varying Width

For flows in rivers and open channels, it can be taken that the flow is dominated along the longitudinal direction, with negligible velocity and free surface variations along lateral and vertical directions [de Boer, 2003]. This important assumption enables the derivation of a more simplified model for such flows. In this section, we derive the set of partial differential equations (PDEs) suitable for dealing with flows in rivers and open channels with arbitrary shaped geometry. These equations, referred to as the Saint Venant Equations or Saint Venant Models (SVM) were first derived in [de Saint-Venant, 1871].

The hypothesis of the SVM include those of the FSEE in section 2.2, in addition to the following :

- The flow is well approximated by a flow with uniform velocity over each cross-section and the free surface is assumed horizontal over each cross section, namely
$$\eta(\vec{\boldsymbol{X}}, t) = \bar{\eta}(x, t). \tag{2.33}$$
- The shallow water assumption in section 2.3 is valid, that is the depth of water is significantly small compared to longitudinal length scales.

We first remark the following: We shall not include friction modelling in the derivations throughout this thesis but shall use the results from literature like those from [Cunge et al., 1980; Stoker, 1957].

2.4.1 Background

There are different approaches to derive these models, such as (i) the infinitesimal elemental control volume approach [Stoker, 1957; de Boer, 2003; Tanguy, 2010; Cunge et al., 1980], (ii) the asymptotic expansion approach [Decoene et al., 2009] and (iii) the direct integration approach [Fernandez-Nieto et al., 2010; Aldrighetti, 2007; Szymkiewicz, 2010]. Here, and throughout this thesis, we shall consistently adopt the direct integration approach whereby we directly integrate the FSEE equations.

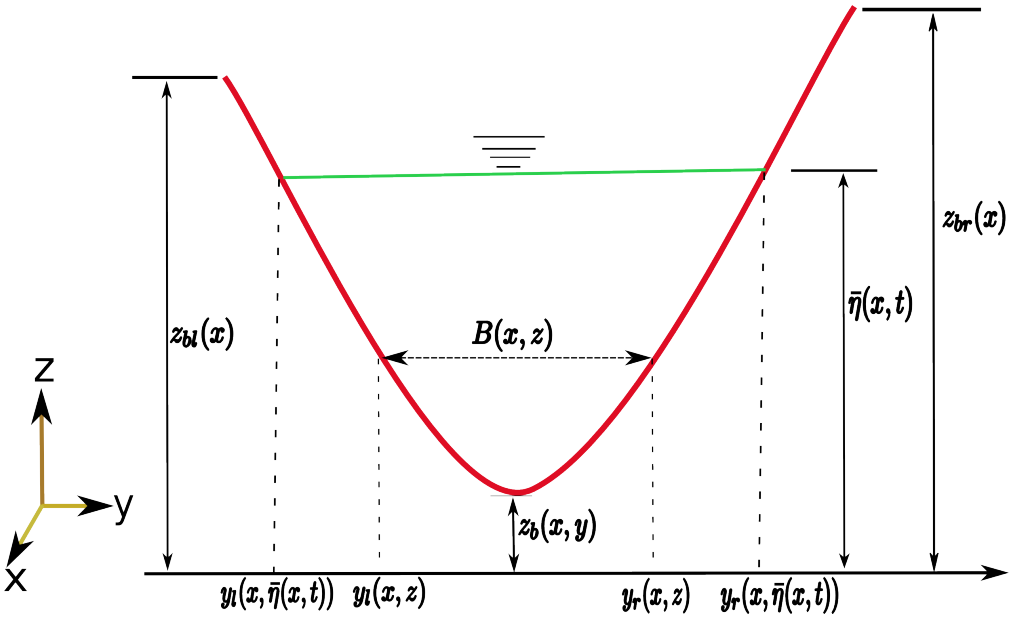


Figure 2.2: Non-full channel flow cross section showing the left and right bank elevations $z_{bl}(x)$ and $z_{br}(x)$, the laterally flat free-surface elevation $\bar{\eta}(x,t)$ (in green), the bottom elevation in 2D $z_b(\vec{\boldsymbol{X}})$ (in red), the width, $B(x,z)$ of the channel at elevation, z above the reference elevation $z=0$, and the $y-$coordinates, $y_l(x,z)$ and $y_r(x,z)$ of the left and right lateral walls at elevation z.

We begin by considering an open channel whose length lies along the x-axis (frontal direction), the width lies along the y-axis (lateral direction) and z-axis is in vertical direction. Figure 2.2 shows a cross section of the channel at point x along its length. We assume that the complete geometry of the channel is known, so the bed elevation, $z_b(x,y)$ is known at all points. The left and right wall/bank elevation, $z_{bl}(x)$ and $z_{br}(x)$ of the channel are also given. $Z_b(x)$ is the bottom elevation of the cross section in 1D sense. Here, we take it to be

$$Z_b(x) = \min_y z_b(x,y).$$

At any elevation, z above the reference $z = 0$, we define the function, $B(x, z)$ to give the lateral width of the channel cross section. Furthermore, we define the functions, $y_l(x, z)$ and $y_r(x, z)$ to give the $y-$coordinates of the left and right lateral boundaries respectively at elevation, z so that

$$B(x, z) = y_r(x, z) - y_l(x, z). \tag{2.34}$$

Note that the width functions; $B(x, z), y_l(x, z), y_r(x, z)$ satisfy the following

$$B(x, z) = 0, y_r(x, z) = y_l(x, z) \text{ for all } z < Z_b(x). \tag{2.35}$$

Now, a very important quantity throughout this thesis, is the maximum channel wall or bank elevation.

Definition 2.4.1 (Maximum channel wall elevation, $z_b^w(x)$)**.** *The maximum channel wall elevation at cross section x, is the minimum elevation of the channel banks above which flooding is said to have occurred. We denote it by $z_b^w(x)$, that is*

$$z_b^w(x) = \min(z_{bl}(x), z_{br}(x)). \tag{2.36}$$

See Figure 2.2.

We shall use the terms maximum channel wall elevation and channel wall elevation interchangeably to refer to $z_b^w(x)$. Note that this quantity is known since the elevation of both banks are already known. In fact, for the cross section depicted in Figure 2.2, $z_b^w(x) = z_{bl}(x)$.

With the channel geometry completely defined, let $\bar{\eta}(x, t)$ be the laterally constant free-surface elevation of fluid in the cross section, see Figure 2.2. Then the water depth in the cross section is given by

$$H(\vec{\boldsymbol{X}}, t) = \bar{\eta}(x, t) - z_b(\vec{\boldsymbol{X}}), \text{ for all } y_l(x, \bar{\eta}(x, t)) \leq y \leq y_r(x, \bar{\eta}(x, t)). \tag{2.37}$$

and the channel flow domain, Ω_{ct} at time, t is given by

$$\Omega_{ct} = \{(\vec{\boldsymbol{X}}, z) : y_l(x, \bar{\eta}(x, t)) \leq y \leq y_r(x, \bar{\eta}(x, t)), z_b(\vec{\boldsymbol{X}}) \leq z \leq \bar{\eta}(x, t))\}. \tag{2.38}$$

Note that the following condition always holds:

$$z_b(\vec{\boldsymbol{X}})|_{y=y_l(x,z),y_r(x,z)} = z \quad \forall z \in [Z_b(x), z_b^w(x)]. \tag{2.39}$$

Since we are concerned with only cases where the channel does not overflow its

banks, namely

$$\bar{\eta}(x,t) \leq z_b^w(x) \quad \forall t \geq 0, \tag{2.40}$$

then by (2.39), we have the following condition to be true:

$$z_b(\vec{\boldsymbol{X}})|_{y=y_l(x,\bar{\eta}(x,t)),y_r(x,\bar{\eta}(x,t))} = \bar{\eta}(x,t) \quad \forall t \geq 0. \tag{2.41}$$

We now proceed to derive the SVM in the following section.

2.4.2 Deriving the SVM

We start with the full 3D Free-Surface Euler Equations, (2.10)-(2.16), and define the following averages :

$$Q(x,t) = \int_{y_l(x,\bar{\eta}(x,t))}^{y_r(x,\bar{\eta}(x,t))} \int_{z_b(\vec{\boldsymbol{X}})}^{\bar{\eta}(x,t)} u(\vec{\boldsymbol{X}},z,t)dzdy, \tag{2.42}$$

$$A(x,t) = \int_{y_l(x,\bar{\eta}(x,t))}^{y_r(x,\bar{\eta}(x,t))} \int_{z_b(\vec{\boldsymbol{X}})}^{\bar{\eta}(x,t)} dzdy = \int_{y_l(x,\bar{\eta}(x,t))}^{y_r(x,\bar{\eta}(x,t))} H(\vec{\boldsymbol{X}},t)dy. \tag{2.43}$$

So that the section-averaged velocity, $\underline{\bar{u}}$ is now given as

$$\underline{u}(x,t) = \frac{Q(x,t)}{A(x,t)} = \frac{1}{A(x,t)} \int_{y_l(x,\bar{\eta}(x,t))}^{y_r(x,\bar{\eta}(x,t))} \int_{z_b(\vec{\boldsymbol{X}})}^{\bar{\eta}(x,t)} u(\vec{\boldsymbol{X}},z,t)dzdy. \tag{2.44}$$

$A(x,t)$ is wetted area of cross section, while $Q(x,t)$ and $\underline{u}$ are the section averaged discharge (volumetric) and velocity respectively, along the longitudinal direction.

First, we note that y-independence of the free-surface, $\bar{\eta}(x,t)$ means that the sum,

$$H(\vec{\boldsymbol{X}},t) + z_b(\vec{\boldsymbol{X}}) = \bar{\eta}(x,t), \quad y_l(x,\bar{\eta}(x,t)) \leq y \leq y_r(x,\bar{\eta}(x,t)) \tag{2.45}$$

is constant in y, even though each of $h(\vec{\boldsymbol{X}},t)$ and $z_b(\vec{\boldsymbol{X}})$ depends on y. See figure 2.2.

We then start the derivation by first noting that the assumption of uniform velocity over the cross section (dominance of flow along longitudinal direction) allows

to neglect the lateral acceleration component, namely

$$\frac{d}{dt}v(\vec{\boldsymbol{X}},z,t) := \partial_t v + \vec{\boldsymbol{U}}\cdot(\nabla_{3D}v) = 0. \tag{2.46}$$

Hence, the y-momentum equation of FSEE, (2.12) reduces to

$$\partial_y P(\vec{\boldsymbol{X}},z,t) = 0, \tag{2.47}$$

$$=> P = P(x,z,t). \tag{2.48}$$

So, pressure does not depend on y. Next, just like in the derivation of the 2D SWE, the assumption of small depth, reduces the z-momentum equation, (2.13) to :

$$P(\vec{\boldsymbol{X}},z,t) = \rho g(\bar{\eta}(x,t) - z), \tag{2.49}$$

$$=> \partial_x P(\vec{\boldsymbol{X}},z,t) = \rho g \partial_x \bar{\eta}(x,t). \tag{2.50}$$

Therefore, the FSEE (2.10) -(2.16), reduce to the following system :

$$\nabla_{3D}\cdot\vec{\boldsymbol{U}} = 0. \tag{2.51}$$

$$\partial_t u + \nabla_{3D}\cdot(u\vec{\boldsymbol{U}}) = -g\partial_x\bar{\eta}(x,t). \tag{2.52}$$

in Ω_{ct}.

$$\Big(\vec{\boldsymbol{u}}\cdot(\nabla z_b(\vec{\boldsymbol{X}})) - w\Big)\Big|_{z=z_b(\vec{\boldsymbol{X}})} = 0. \tag{2.53}$$

$$(\partial_t\bar{\eta}(x,t) + u\partial_x\bar{\eta}(x,t) - w)\Big|_{z=\bar{\eta}(x,t)} = 0. \tag{2.54}$$

The Mass Conservation Equation

Integrating equation (2.51) vertically and laterally over the cross section, gives

$$\int_{y_l(x,\bar{\eta}(x,t))}^{y_r(x,\bar{\eta}(x,t))}\int_{z_b(\vec{\boldsymbol{X}})}^{\bar{\eta}(x,t)} (\partial_x u + \partial_y v + \partial_z w)\,dzdy = 0. \tag{2.55}$$

For convenience, we drop the notations for the independent variables. Using the Leibniz rule, the inner integral leads to

$$
\begin{aligned}
0 &= \int_{y_l}^{y_r} \left\{ \partial_x \int_{z_b}^{\bar{\eta}} u\, dz + \partial_y \int_{z_b}^{\bar{\eta}} v\, dz + \Big[w - u\partial_x \bar{\eta} \Big]_{z=\bar{\eta}} + \Big[\vec{\boldsymbol{u}} \cdot (\nabla z_b) - w \Big]_{z=z_b} \right\} dy \\
&= \int_{y_l}^{y_r} \left\{ \partial_x \int_{z_b}^{\bar{\eta}} u\, dz \right\} dy + \int_{y_l}^{y_r} \left\{ \partial_y \int_{z_b}^{\bar{\eta}} v\, dz \right\} dy + \int_{y_l}^{y_r} \partial_t H(\vec{\boldsymbol{X}}, t)\, dy \quad \text{by (2.53) } - \text{ (2.54).}
\end{aligned}
$$

Applying the Leibniz rule again, we have

$$
\begin{aligned}
0 = \partial_x \int_{y_l}^{y_r} \int_{z_b}^{\bar{\eta}} u\, dz dy &- \Big[\int_{z_b}^{\bar{\eta}} u\, dz \Big]_{y=y_r} \partial_x y_r + \Big[\int_{z_b}^{\bar{\eta}} u\, dz \Big]_{y=y_l} \partial_x y_l + \Big[\int_{z_b}^{\bar{\eta}} v\, dz \Big]_{y=y_r} \\
&- \Big[\int_{z_b}^{\bar{\eta}} v\, dz \Big]_{y=y_l} + \partial_t \int_{y_l}^{y_r} H(\vec{\boldsymbol{X}}, t)\, dy.
\end{aligned}
$$

Using the definitions in (2.42) and (2.43), we have the following equation:

$$
\partial_t A(x,t) + \partial_x Q(x,t) = \Phi^A(x,t). \tag{2.56}
$$

where

$$
\begin{aligned}
\Phi^A(x,t) = &\Big[\int_{z_b}^{\bar{\eta}} u\, dz \Big]_{y=y_r(x,\bar{\eta})} \partial_x y_r(x,\bar{\eta}) - \Big[\int_{z_b}^{\bar{\eta}} u\, dz \Big]_{y=y_l(x,\bar{\eta})} \partial_x y_l(x,\bar{\eta}) \\
&- \Big[\int_{z_b}^{\bar{\eta}} v\, dz \Big]_{y=y_r(x,\bar{\eta})} + \Big[\int_{z_b}^{\bar{\eta}} v\, dz \Big]_{y=y_l(x,\bar{\eta})}.
\end{aligned} \tag{2.57}
$$

Since we assume that the channel does not overflow, all the integrals in Φ^A are zero because their limits are equal (see equation, (2.41)), hence (2.56) leads to the following mass conservation equation:

$$
\partial_t A(x,t) + \partial_x Q(x,t) = 0. \tag{2.58}
$$

The Momentum Equation

First, we note that the right hand side of equation (2.52) is independent of y and z, which means that the left hand side, $\frac{d}{dt} u(\vec{\boldsymbol{X}}, z, t)$ is also independent of y and z,

hence u is independent of y and z. This allows to expand $u(\vec{\boldsymbol{X}}, z, t)$ as follows :

$$u(\vec{\boldsymbol{X}}, z, t) = \underline{u}(x, t) + \widetilde{\underline{u}}(\vec{\boldsymbol{X}}, z, t), \tag{2.59}$$

where

$$\widetilde{\underline{u}}(\vec{\boldsymbol{X}}, z, t) \ll 1, \text{ such that } \widetilde{\underline{u}}^k(\vec{\boldsymbol{X}}, z, t) = 0 \quad \forall k \geq 2 \tag{2.60}$$

and

$$\overline{\widetilde{\underline{u}}(\vec{\boldsymbol{X}}, z, t)} := \frac{1}{A(\vec{\boldsymbol{X}}, z, t)} \int\limits_{y_l(x,\bar{\eta}(x,t))}^{y_r(x,\bar{\eta}(x,t))} \int\limits_{z_b(\vec{\boldsymbol{X}})}^{\bar{\eta}(x,t)} \widetilde{\underline{u}}(\vec{\boldsymbol{X}}, z, t) dz dy = 0. \tag{2.61}$$

Integrating equation (2.52) vertically and laterally over the cross section, we have

$$\int\limits_{y_l}^{y_r} \left\{ \int\limits_{z_b}^{\bar{\eta}} \Big(\partial_t u + \partial_x(u^2) + \partial_y(uv) + \partial_z(uw) \Big) dz \right\} dy = -g\partial_x\bar{\eta} \int\limits_{y_l}^{y_r} \int\limits_{z_b}^{\bar{\eta}} dz dy. \tag{2.62}$$

That is,

$$\begin{aligned} -gA\partial_x\bar{\eta} = \int\limits_{y_l}^{y_r} \Bigg(\partial_t \int\limits_{z_b}^{\bar{\eta}} u\, dz + \partial_x \int\limits_{z_b}^{\bar{\eta}} u^2\, dz + \partial_y \int\limits_{z_b}^{\bar{\eta}} uv\, dz + \Big[u(w - u\partial_x\bar{\eta} - \partial_t\bar{\eta}) \Big]_{z=\bar{\eta}} \\ + \Big[u(\vec{\boldsymbol{u}} \cdot (\nabla z_b) - w) \Big]_{z=z_b} \Bigg) dy, \quad \text{by Leibniz rule.} \end{aligned}$$

Using the kinematic boundary conditions (2.53)-(2.54), we have

$$\begin{aligned} -gA\partial_x\bar{\eta} &= \int\limits_{y_l}^{y_r} \Big(\partial_t \int\limits_{z_b}^{\bar{\eta}} u\, dz \Big) dy + \int\limits_{y_l}^{y_r} \Big(\partial_x \int\limits_{z_b}^{\bar{\eta}} u^2\, dz \Big) dy + \int\limits_{y_l}^{y_r} \Big(\partial_y \int\limits_{z_b}^{\bar{\eta}} uv\, dz \Big) dy \\ &= \partial_t \int\limits_{y_l}^{y_r} \int\limits_{z_b}^{\bar{\eta}} u\, dz dy + \partial_x \int\limits_{y_l}^{y_r} \int\limits_{z_b}^{\bar{\eta}} u^2\, dz dy - \Big[\int\limits_{z_b}^{\bar{\eta}} u^2 dz \Big]_{y=y_r} \partial_x y_r \\ &\quad + \Big[\int\limits_{z_b}^{\bar{\eta}} u^2 dz \Big]_{y=y_l} \partial_x y_l + \Big[\int\limits_{z_b}^{\bar{\eta}} uv dz \Big]_{y=y_r} \partial_x y_r - \Big[\int\limits_{z_b}^{\bar{\eta}} uv dz \Big]_{y=y_l} \partial_x y_l. \end{aligned}$$

Using equations (2.59)-(2.61) to simplify the second term on the right of the last

equality sign, we have the following equation:

$$-gA\partial_x\bar{\eta} = \partial_t \int_{y_l}^{y_r}\int_{z_b}^{\bar{\eta}} u\,dzdy + \partial_x\left(\underline{u}^2(x,t)\int_{y_l}^{y_r}\int_{z_b}^{\bar{\eta}} dzdy\right) - \Phi^Q(x,t), \tag{2.63}$$

where

$$\begin{aligned}\Phi^Q(x,t) = &\left[\int_{z_b}^{\bar{\eta}} u^2dz\right]_{y=y_r}\partial_x y_r - \left[\int_{z_b}^{\bar{\eta}} u^2dz\right]_{y=y_l}\partial_x y_l \\ &- \left[\int_{z_b}^{\bar{\eta}} uvdz\right]_{y=y_r}\partial_x y_r + \left[\int_{z_b}^{\bar{\eta}} uvdz\right]_{y=y_l}\partial_x y_l.\end{aligned} \tag{2.64}$$

Using the definitions of A and Q, (2.63) gives the following important equation

$$\partial_t Q(x,t) + \partial_x\left(\frac{Q^2((x,t))}{A(x,t)}\right) = -gA(x,t)\partial_x\bar{\eta} + \Phi^Q(x,t). \tag{2.65}$$

Using equation (2.41), all the integrals on the right vanish. Then, we obtain the following equation for the momentum.

$$\partial_t Q(x,t) + \partial_x\left(\frac{Q^2(x,t)}{A(x,t)}\right) = -gA(x,t)\partial_x\bar{\eta}(x,t). \tag{2.66}$$

Summary of the Models

In summary, the Saint Venant Model (SVM) for arbitrary cross sectional open channel flow is the following:

$$\begin{aligned}\partial_t A(x,t) + \partial_x Q(x,t) &= 0. \\ \partial_t Q(x,t) + \partial_x\left(\frac{Q^2(x,t)}{A(x,t)}\right) &= -gA(x,t)\partial_x\bar{\eta}(x,t).\end{aligned} \tag{2.67}$$

We make the following remarks.

Remark 2.4.1. *Observe that the model,* (2.67) *has two equations but three unknowns; $A, Q, \bar{\eta}(x,t)$. Given A, Q and the channel geometry, one can calculate the third variable, $\bar{\eta}(x,t)$. With the given wetted cross sectional area A, the channel geometry provides the information (function), $\mathcal{H}(A, Z_b)$ which computes the water depth, $\underline{H}$, measured from say $Z_b(x)$ to the unknown flat free surface. Then the unknown flat free-surface is computed as $\bar{\eta}(x,t) = Z_b + \underline{H}(x)$. Hence every channel has its own definition of the function, $\mathcal{H}(A, Z_b)$. For instance, rectangular channel*

has $\mathcal{H}(A, Z_b) = A/B(x)$, where $B(x)$ is the channel width known from the geometry. We make the function depend on Z_b to indicate the point from which the water height is measured. This function is extensively used in the vertical coupling method in chapter 5.

Remark 2.4.2. *We choose to keep the model,* (2.67) *in this form, instead of divergence form, because different numerical schemes adopt different (like divergence or quasi-linear) forms of the model (see [Cunge et al., 1980]). Moreover, we do not intend to involve in theoretical aspect of this model, rather, we use existing methods for solving them.*

2.4.3 Locally Rectangular Channel

For a channel with rectangular cross section, we have the following simplifications: $B = B(x), z_b(x, y) = Z_b(x), H(\vec{\boldsymbol{X}}, t) = \underline{H}(x, t) = \bar{\eta}(x, t) - Z_b(x), A(x, t) = \underline{H}(x, t)B(x)$. Therefore, the SVM (2.67) becomes:

$$\partial_t A(x,t) + \partial_x Q(x,t) = 0. \tag{2.68}$$

$$\partial_t Q(x,t) + \partial_x \left(\frac{Q^2(x,t)}{A(x,t)} + g\frac{A^2(x,t)}{2B(x)} \right) = g\frac{A^2(x,t)}{2B^2(x)}\partial_x B(x) - gA(x,t)\partial_x Z_b(x). \tag{2.69}$$

2.4.4 1D Shallow Water Equations

The 1D SVM reduce to the 1D Shallow water equations if the channel lateral width is constant. That is, if

$$B = \text{ constant}$$

then, the models, (2.68) and (2.69) reduce to

$$\begin{aligned} \partial_t \underline{H}(x,t) + \partial_x \bar{q}(x,t) &= 0, \\ \partial_t \bar{q}(x,t) + \partial_x \left(\frac{\bar{q}^2(x,t)}{\underline{H}(x,t)} + \frac{g}{2}\underline{H}^2(x,t) \right) &= -g\underline{H}(x,t)\partial_x Z_b(x), \end{aligned} \tag{2.70}$$

where $\bar{q}(x,t) = \underline{H}(x,t)\underline{u}(x,t)$.

2.5 Properties of Hyperbolic Conservation Laws

As we shall see in this section, the models derived in the previous sections fall into the class of PDEs called hyperbolic PDEs, in particular, hyperbolic conservation laws with source terms or hyperbolic balance laws or inhomogeneous hyperbolic

conservation laws. Most of the numerical methods for this kind of equations are usually based on the methods originally developed for their corresponding homogeneous conservation laws. These methods for conservation laws are formulated in such a way as to overcome most of the difficulties associated with their mathematical properties. These mathematical properties are what we briefly present in this section.

This section starts with a brief explanation of what a hyperbolic PDE is, then goes ahead to summarily derive and present conservation laws. Next, we use the eigen-structure of our models to show that they are hyperbolic conservation laws with source terms. Finally, we briefly discuss some of those mathematical properties which necessitate the type of numerical methods usually deployed to solve them.

2.5.1 Hyperbolic PDEs

Consider the following system of M quasi-linear PDEs in N-dimensional space :

$$\partial_t w + \sum_{i=1}^{N} A_i(w)\partial_{x_i} w = 0, \tag{2.71}$$

where $w(\vec{x}, t) : \mathbb{R}^N \times \mathbb{R}^+ \rightarrow \mathbb{R}^M$,
$A_i = A_i(\vec{x}, t, w) \in \mathbb{R}^M \times \mathbb{R}^M, \vec{x} \in \mathbb{R}^N$ and $t \in \mathbb{R}^+$.

The system (2.71) is called hyperbolic if for any unit vector, $\vec{e} = \{e_1, e_2, ..., e_N\} \in \mathbb{R}^N$, the matrix,

$$\sum_{i=1}^{N} A_i e_i$$

has real eigenvalues $\lambda_j(\vec{x}, t, w, e), j = 1, .., M$. It is called strictly hyperbolic if all eigenvalues are real and distinct.

2.5.2 Hyperbolic Conservation Laws

This subsection follows the presentation in [Lax, 1973]. A conservation law states that the rate of change of total amount, $\int_\Omega w(\vec{x}, t)d\vec{x}$ of a quantity contained in a domain Ω, is equal to the flux, $\vec{H}$ across the boundaries of Ω, provided the quantity is neither created nor destroyed within Ω. That is

$$\frac{d}{dt}\left(\int_\Omega w(\vec{x}, t)d\vec{x}\right) = -\int_{\partial\Omega} \vec{H} \cdot \vec{n} dS, \tag{2.72}$$

where w is the density of the quantity in Ω, $\vec{n}$ is outward normal to Ω and dS is the surface element on boundary, $\partial\Omega$ of Ω.

Use of divergence theorem on the right term and the Leibniz formula on the left term gives the following integral form of the conservation law :

$$\int_{\Omega} \left(\partial_t w(\vec{x}, t) + \nabla \cdot \vec{H} \right) d\vec{x} = 0. \tag{2.73}$$

Since Ω is arbitrary, assuming that all partial derivatives of w and $\vec{H}$ are continuous, then the above equation gives the following differential form of the conservation law:

$$\partial_t w(\vec{x}, t) + \nabla \cdot \vec{H} = 0. \tag{2.74}$$

Equations (2.73) and (2.74) are the integral and differential forms respectively, of a scalar conservation law.

Similarly, a system of M conservation laws in differential form is

$$\partial_t w^j(\vec{x}, t) + \nabla . \vec{H}^j = 0, \quad \forall j = 1, .., M \tag{2.75}$$

where $\vec{H}^j = \vec{H}^j(w) = (H_1^j(w), H_2^j(w), ..., H_N^j(w))^T, j = 1, ..M$ is the flux vector in N-dimensions.

$$\partial_t w^j(\vec{x}, t) + \partial_{x_1} H_1^j + \partial_{x_2} H_2^j + ... + \partial_{x_N} H_N^j = 0, \quad \forall j = 1, .., M$$

or

$$\partial_t w(\vec{x}, t) + \partial_{x_1} \begin{pmatrix} H_1^1 \\ ... \\ H_1^M \end{pmatrix} + \cdots + \partial_{x_N} \begin{pmatrix} H_N^1 \\ ... \\ H_N^M \end{pmatrix} = 0 \text{ or } \partial_t w(\vec{x}, t) + \sum_{i=1}^{N} \partial_{x_i} H_i(w) = 0,$$

where $\vec{H}_i(w) = (H_i^1(w), ..., H_i^M(w))^T, i = 1, .., N$, so the system of conservation laws in quasi-linear form becomes

$$\partial_t w(\vec{x}, t) + \sum_{i=1}^{N} A_i(w) \partial_{x_i} w = 0, \tag{2.76}$$

where $A_i(w) = \partial_w H_i$. Therefore by the definition of hyperbolicity in section 2.5.1, the system of conservation laws, (2.75) is said to be hyperbolic if the quasi-linear form, (2.76) is hyperbolic. If the flux function, $\vec{H}(w) = (H_1(w), ..., H_N(w))$ is non-linear in w, then it is a non-linear system of hyperbolic conservation laws. In

addition, if a non-zero right hand side, say S_c, is included, then we call it hyperbolic system of conservation laws with source terms or hyperbolic balance laws.

2.5.3 Hyperbolicity of the Derived Models

Here, we quickly show that the 2D SWE, (2.29)-(2.31) and the 1D SVM, (2.68) and (2.69) are non-linear hyperbolic conservation laws with source terms. To this end, the two systems can generally be written as

$$\partial_t w + \nabla \cdot \vec{H}(w) = S_c \tag{2.77}$$

where the variables are defined as follows :

- For the 2D SWE, (2.32), $w = \Pi$, the flux is $\vec{H} = F(\Pi)$ and the source is $S_c = S(\Pi, z_b)$ where Π, F, S are given in (2.32). Hence, the Jacobian is

$$\begin{aligned} J(\Pi) &= \omega_1 \partial_\Pi F_1(\Pi) + \omega_2 \partial_\Pi F_2(\Pi) \\ &= \begin{pmatrix} 0 & \omega_1 & \omega_2 \\ (c^2 - \bar{u}^2)\omega_1 - \bar{u}\bar{v}\omega_2 & 2\bar{u}\omega_1 + \bar{v}\omega_2 & \bar{u}\omega_2 \\ -\bar{u}\bar{v}\omega_1 + (c^2 - \bar{v}^2)\omega_2 & \bar{v}\omega_1 & \bar{u}\omega_1 + 2\bar{v}\omega_2 \end{pmatrix} \end{aligned}$$

  whose eigenvalues are $\lambda_1 = \bar{u}\omega_1 + \bar{v}\omega_2 - c|\vec{w}|$, $\lambda_2 = \bar{u}\omega_1 + \bar{v}\omega_2$ and $\lambda_3 = \bar{u}\omega_1 + \bar{v}\omega_2 + c|\vec{w}|$ (see [Toro, 2001]). Where $\bar{u} = q_x/H, \bar{v} = q_y/H$, $c = \sqrt{gH}$ and $\vec{\omega} = \{\omega_1, \omega_2\} \in \mathbb{R}^2$. Since $\vec{\omega}$ is real, all eigenvalues are real, so the system is hyperbolic. If $H > 0$, then the system is strictly hyperbolic since all eigenvalues are distinct then.

- For the 1D SVM, (2.68) and (2.69),
  $w = (A, Q)^T$, the flux is $\vec{H}(w) = \vec{¶}(w)$ where $\vec{¶}(w) = \left(Q, \frac{Q^2}{A} + g\frac{A^2}{2B}\right)^T$, the source term is $S(w) = \left(0, g\frac{A^2}{B^2}\partial_x B - gA\partial_x z_b\right)^T$.
  And the Jacobian is

$$J(w) = \partial_w \vec{¶}(w) = \begin{pmatrix} 0 & 1 \\ c^2 - \underline{u}^2 & 2\underline{u} \end{pmatrix}$$

  whose eigenvalues are $\lambda_1 = \underline{u} - c, \lambda_2 = \underline{u} + c$ (see [Aldrighetti, 2007]) which are all real, so the system is hyperbolic. Here, $\underline{u} = Q/A$ and $c = \sqrt{gA/B} = \sqrt{g\underline{H}}$. Similar to the 2D system above, if $\underline{H} = A/B > 0$, then the system is strictly hyperbolic.

2.5.4 Properties of hyperbolic PDEs

We now briefly highlight two basic properties of hyperbolic PDEs which necessitate the choice of solution methods usually deployed for non-linear hyperbolic conservation laws.

Weak or Discontinuous Solutions

It is well known that solutions of non-linear hyperbolic conservation laws can develop discontinuities at finite time, even when initialised with smooth data [Bressan, 2009; Lax, 1973; Toro, 2001; Renardy and Rogers, 2004]. In such cases, the differential forms of the conservation laws are no longer valid since derivatives no longer exist in the classical sense. However, as pointed out in Lax [1973], since a conservation law is an integral relation, it may be satisfied by functions which are not diffentiable, not even continuous. This means that conservation laws can have discontinuous solutions, generally called weak solutions.

Therefore, to understand and construct global solution, in time, for non-linear conservation laws, one needs to extend the notion of solutions to include discontinuous functions. Define the space :

$$\begin{aligned}\mathcal{C}_0^1(\Omega \times \mathbb{R}^+) = \big\{ f : \Omega \times \mathbb{R}^+ \to \mathcal{R} : f \in C^1(\Omega \times \mathbb{R}^+),\ \text{supp } f \subset K, \\ K \text{ compact in } \Omega \times [0, \infty) \big\}, \\ \text{supp } f = \overline{\{(\vec{\boldsymbol{X}}, t) : f(\vec{\boldsymbol{X}}, t) \neq 0\}}.\end{aligned}$$

If we multiply the conservation law (2.74) by a test function $\psi(\vec{\boldsymbol{X}}, t) \in \mathcal{C}_0^1(\Omega \times \mathbb{R}^+)$ and integrate over $\Omega \times \mathcal{R}^+$, we have

$$\int_0^\infty \int_\Omega \left(\psi(\vec{\boldsymbol{X}}, t) \partial_t w(\vec{\boldsymbol{X}}, t) + \psi(\vec{\boldsymbol{X}}, t) \nabla \cdot \vec{H} \right) d\vec{\boldsymbol{X}} dt = 0. \tag{2.78}$$

Evaluating the integral, using integration by parts and noting that the test function, $\psi(\vec{\boldsymbol{X}}, t)$ vanishes at boundaries, we have

$$\begin{aligned}\int_0^\infty \int_\Omega \left(w(\vec{\boldsymbol{X}}, t) \partial_t \psi(\vec{\boldsymbol{X}}, t) + \vec{H} \nabla \psi(\vec{\boldsymbol{X}}, t) \right) d\vec{\boldsymbol{X}} dt = - \int_\Omega \psi(\vec{\boldsymbol{X}}, 0) w_0(\vec{\boldsymbol{X}}) d\vec{\boldsymbol{X}} \\ \forall \quad \psi(\vec{\boldsymbol{X}}, t) \in \mathcal{C}_0^1(\Omega \times \mathbb{R}^+)\end{aligned} \tag{2.79}$$

and $w_0(\vec{\boldsymbol{X}})$ is the initial data.

A function, $w(\vec{\boldsymbol{X}}, t)$ is said to be a weak solution to the the conservation law, (2.74) if (2.79) holds. So, the definition of the weak solution does not require the differentiability of the solution $w(\vec{\boldsymbol{X}}, t)$ or the flux function, $\vec{H}$. If a function $w(\vec{\boldsymbol{X}}, t)$ is a classical solution of the conservation law (2.74), then it is also a weak solution of the problem. The proof of this and other details on weak solutions can be found in texts such as [Renardy and Rogers, 2004; Evans, 2010] .

Non-Uniqueness of Weak Solutions

Weak solutions to non-linear hyperbolic conservation laws are known to be non-unique [Leveque, 1992; Lax, 1973; Renardy and Rogers, 2004]). Since it is our hope that conservation laws model real life, then we expect them to have physically relevant weak solutions. The task then is how to pick out this physically correct weak solution. It is frequently possible to derive conditions that can be directly imposed on weak solutions of hyperbolic conservation laws to select the physically correct weak solution [Leveque, 1992]. These conditions are called entropy conditions. Therefore, the desired physically correct weak solution is one that satisfies an entropy condition. We recall the following definitions (see [Bouchut and de Luna, 2009; Harten, 1983; Vázquez-Cendón, 2015; Mungkasi, 2012] for these definitions). Consider the following system of conservation laws.

$$\partial_t w + \partial_x f(w) = 0, \quad x \in [x_a, x_b] \subset \mathbb{R} \quad t \geq 0 \tag{2.80}$$

where $w(x,t) \in \mathbb{R}^M$, $f \in \mathbb{R}^M, M \in \mathbb{Z}^+$.

Definition 2.5.1. *An entropy for a system* (2.80) *is a real-valued convex function,* $\eta(w)$ *of* w *(i.e.* $\partial^2_{ww}\eta(w) > 0$*) such that there exists another real-valued function, called the entropy flux,* $g(w)$ *satisfying the following:*

$$\partial_w g(w) = \partial_w \eta(w) \partial_w f(w), \tag{2.81}$$

and the pair (η, g) *is called the entropy-entropy flux pair.*

Definition 2.5.2. *An entropy satisfying weak solution of the conservation law,* (2.80) *is one which satisfies the following inequality,in the weak sense,*

$$\partial_t \eta(w) + \partial_x g(w) \leq 0 \tag{2.82}$$

for any entropy-entropy flux pair, (η, g)*.*

The inequality (2.82) becomes an equality, called the entropy equation, if

the solution is smooth. It is strictly inequality, in the weak sense, if the solution is discontinuous [Mungkasi, 2012]. This concept of entropy inequality have been explored for grid adaptivity in [Puppo, 2002, 2004; Mungkasi, 2012] among many others. Based on the above discussion, it is then important that methods for solving non-linear hyperbolic conservation laws must be able to compute discontinuous (weak) solutions and also be able to automatically choose the physically correct entropy satisfying weak solution. In the following chapter, we give an overview of the methods adopted in this thesis for the numerical approximation of the solutions of the models derived in this chapter.

Chapter 3

Finite Volume Schemes For Free-Surface Flow Problems

3.1 Introduction

The goal of this chapter is to give an overview of some of the methods for approximating the solutions of the channel and flood models derived in the previous chapter and use the results to justify some of the basic assumptions in coupled 2D/1D simulations. As discussed in the previous chapter, these models are hyperbolic conservation laws with source terms and the methods (numerical) for solving them are based on those developed for conservation laws without source terms. Hence we begin the chapter by introducing some of the concepts and methods for approximating the solution of systems of conservation laws in 1D. In this introductory section, we restate the 1D system of conservation laws, introduce the 1D computational domain, called the grid or mesh, and explain why the cell average of the true solution is what we wish to approximate. Thereafter, the outline of the chapter is given.

Let us start by considering the following $1D$ system of $M \in \mathbb{N}$ conservation laws,

$$\partial_t \boldsymbol{W} + \partial_x f(\boldsymbol{W}) = 0, \quad x \in [x_a, x_b] \subset \mathbb{R} \quad t \geq 0 \tag{3.1}$$

where $\boldsymbol{W}(x,t) \in \mathbb{R}^M$, $f \in \mathbb{R}^M$ and the system is closed with appropriate boundary and initial conditions.

Let $\{x_{i+1/2}\}_{i=0}^{N}$ be set of $N+1 \in \mathbb{N}$ points in the domain such that $x_{-1/2} = x_a$ and $x_{N+1/2} = x_b$. Then we discretize the domain $[x_a, x_b]$ into $N+1$ control volumes or cells, $K_i = [x_{i-1/2}, x_{i+1/2}]$ with center, $x_i = \frac{x_{i-1/2}+x_{i+1/2}}{2}$ and width, $\Delta x_i = x_{i+1/2} - x_{i-1/2}$, see figure 3.1. Let $n \in \mathbb{N}$ and $t^n \in \mathbb{R}$ be a given time, the discrete time step size, Δt^n is calculated from the solution at time, t^n. The process

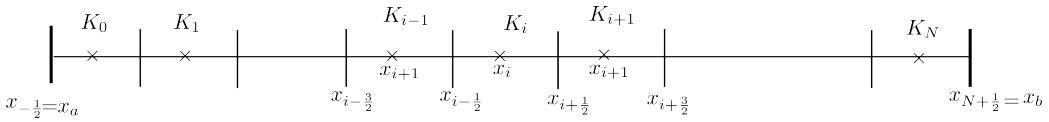


Figure 3.1: Discretization of the 1D domain $[x_a, x_b] \subset \mathbb{R}$ into $N+1$ grid cells. $K_i = [x_{i-1/2}, x_{i+1/2}]$ is the i-th cell, for $i = 0, 1, 2, ..., N$. $x_i = \frac{x_{i-1/2}+x_{i+1/2}}{2}$ and $\Delta x_i = x_{i+1/2} - x_{i-1/2}$ are center and width of K_i respectively.

of computing time step size, Δt^n shall be postponed to section 3.7. Here, we assume that it is given and is constant in n, that is $\Delta t^n = \Delta t$.

The values,

$$\overline{\boldsymbol{W}}_i^n := \frac{1}{\Delta x_i} \int_{K_i} \boldsymbol{W}(x, t^n) dx, \quad i = 0, 1, 2, ..., N$$

are the averages of the true solution in the cells, $i = 0, 1, 2, ..., N$. As noted in the previous chapter, the true solution of the conservation law, (3.1) is not generally continuous, so point-wise values, $\boldsymbol{W}(x_i, t^n)$, of the true solution may not make sense, hence we do not aim to approximate these point-wise values. On the other hand, the averages, $\overline{\boldsymbol{W}}_i^n$ of the true solution continue to make sense even when the true solution is discontinuous. Therefore, the numerical solution, $\boldsymbol{W}_i^n$ we seek, would be the approximate values of these cell averages, namely

$$\boldsymbol{W}_i^n \approx \frac{1}{\Delta x_i} \int_{K_i} \boldsymbol{W}(x, t^n) dx. \tag{3.2}$$

So, an approximate solution of the conservation law, (3.1) actually means an approximation of cell averages, not an approximation of point values.

The rest of the chapter is organised as follows. In section 3.2, we briefly discuss the concept of conservative numerical methods, providing the framework for the numerical solution of systems such as (3.1) and we briefly explain why we use the class of conservative methods, called the finite volume methods, throughout this thesis. In section 3.3, the finite volume methods. The finite volume methods for the 2D flood model, (2.32) and the 1D SVM, (2.67) are given in sections 3.4 and 3.5 respectively. In sections 3.6 and 3.7, the numerical boundary conditions and how we compute the time step sizes for all our simulations, are discussed. The chapter concludes in section 3.8 where numerical results are presented and used to demonstrate (i) that the methods are properly implemented and (ii) to numerically verify (a) that channel flow is 1D if not flooding and (b) that there are 2D flow

structure in the channel if flooding. This motivates the methods proposed in the subsequent chapters.

3.2 Conservative Numerical Schemes

As discussed in the previous chapter, it is known that hyperbolic conservation laws exhibit discontinuous solutions even when initialised with smooth data. This leads to the notion of weak solutions which are known to be non-unique. Therefore, it is necessary to consider the class of numerical methods that are capable of computing the physically correct weak solutions of conservation laws. Numerical methods used to compute discontinuous solutions may face two well known issues namely : (i) The solution may converge to a function that is not even a weak solution [Hou and Le Floch, 1994; Durran, 2010; Leveque, 1992; Toro, 2001]. This is because different conservation laws might have the same smooth solution but have different weak solutions. See an example in chapter 3.7 in [Leveque, 1992]. (ii) The solution may converge to the wrong weak solution. The reason is, that weak solutions are not unique means that a method may not necessarily converge to the physically meaningful weak solution. In this section, we briefly discuss the numerical methods, from the literature, which overcome these issues.

A **conservative numerical method** for approximating the solution of (3.1) is a numerical scheme of the form,

$$\boldsymbol{W}_i^{n+1} = \boldsymbol{W}_i^n - \frac{\Delta t_n}{\Delta x_i}\left(\phi_{i+1/2}^n - \phi_{i-1/2}^n\right) \tag{3.3}$$

where

$$\phi_{i+1/2}^n = \phi(\boldsymbol{W}_{i-s}^n, \boldsymbol{W}_{i-s+1}^n, ..., \boldsymbol{W}_{i+k-1}^n, \boldsymbol{W}_{i+k}^n)$$

and ϕ, which has $s+k+1$ arguments, is called the numerical flux function and approximates the analytical flux, $f(\boldsymbol{W})$. $s, k \geq 0$ are some integers. For first-order methods, $s = 0, k = 1$, hence

$$\phi_{i+1/2}^n = \phi(\boldsymbol{W}_i^n, \boldsymbol{W}_{i+1}^n).$$

We shall consider only first-order methods throughout this thesis. It is required that the numerical flux, ϕ be consistent with the analytical flux function, $f(\boldsymbol{W})$, that is

$$\phi(v, v, ..., v) = f(v) \qquad \text{for any constant, } v \tag{3.4}$$

If ϕ is consistent with f, then the numerical method, (3.3) is said to be **consistent**

with the conservation law, (3.1). Finally, ϕ is required to be Lipschitz continuous, namely that there exists a constant $\lambda \geq 0$ such that

$$|\phi(v_1, v_2, ..., v_{s+k+1}) - f(\bar{v})| \leq \lambda \max_i |v_i - \bar{v}| \tag{3.5}$$

for all v_i sufficiently close to $\bar{v}$ [Leveque, 1992].

The problem of numerical solutions converging to functions that are not weak solutions, is then resolved by the theorem of [Lax and Wendroff, 1960] which proved that any sequence of numerical approximations computed with a consistent and conservative numerical method, if convergent, would converge to a weak solution of the conservation law. It is important to note the following facts about the Lax-Wendroff theorem quoted above.

- The theorem did not say that the method is convergent. It only says that if the method is convergent, then it converges to a weak solution. So, the convergence requirement is not address by the theorem neither is it pursued in this thesis.

- The theorem did not say that the method, if convergent, would converge to the correct physically relevant solution. It only said that the solution would converge to a weak solution which may be the correct or wrong weak solution. So, the theorem does not resolve the problem of uniqueness.

- As pointed out in [Toro, 2001], the theorem did not say that only conservative methods, if convergent, converge to weak solutions. However, as also pointed out in [Toro, 2001], it has been shown in [Hou and Le Floch, 1994], for $1D$ scalar conservation laws, that numerical solutions computed by non-conservative methods, if convergent, would converge to the solution of a new conservation law with source term which only vanishes where the solution is smooth.

Therefore, the problem of uniqueness of computed weak solution is not addressed by the theorem of [Lax and Wendroff, 1960] nor is it discussed in this thesis. More on this topic can be found in [Bouchut and de Luna, 2009; Harten, 1983; Vázquez-Cendón, 2015; Leveque, 1992, 2002].

The above discussion leaves us with the choice of using either the finite volume methods or the Discontinuous Galerkin (DG) finite element methods. Both of these methods ensure local conservation which the continuous finite element method (FEM) does not, and they also allow for general unstructured grids which the finite difference method does not. However, the finite volume methods are simple to

implement than the DG methods. Moreover, they are the bases for the local conservation of the DG methods which are more suitable for high order methods. We only discus first order methods in this thesis. In addition, the literature on coupling methods for the 2D-1D shallow water equations is hugely based on the finite volume methods. Therefore, we adopt the finite volume method throughout this thesis. In the next section, we introduce the the finite volume methods.

3.3 Finite Volume Methods for Conservation Laws

The finite volume methods are a class of conservative methods and they approximate the cell average values of the true solution of unknown at each time step. In this section, we will derive the finite volume scheme for the conservation law, (3.1). The finite volume method is based on the integral formulation of the conservation law as we present in the following. Integrating the conservation law, (3.1) over the space-time domain, $[x_{i-1/2}, x_{i+1/2}] \times [t^n, t^{n+1}]$, we have

$$\int_{t^n}^{t^{n+1}} \int_{x_{i-1/2}}^{x_{i+1/2}} \partial_t \boldsymbol{W} dx dt + \int_{t^n}^{t^{n+1}} \int_{x_{i-1/2}}^{x_{i+1/2}} \partial_x f(\boldsymbol{W}) dx dt = 0.$$

Since the limits of integration are constants, we can interchange integrals. So by the fundamental theorem of calculus, we have

$$\int_{x_{i-1/2}}^{x_{i+1/2}} \boldsymbol{W}(x, t^{n+1}) dx - \int_{x_{i-1/2}}^{x_{i+1/2}} \boldsymbol{W}(x, t^n) dx = - \int_{t^n}^{t^{n+1}} f(\boldsymbol{W}(x_{i+1/2}, t)) dt + \int_{t^n}^{t^{n+1}} f(\boldsymbol{W}(x_{i-1/2}, t)) dt.$$

Dividing through by Δx_i, using the definition of w_i^n in (3.2), we have

$$\boldsymbol{W}_i^{n+1} = \boldsymbol{W}_i^n - \frac{\Delta t}{\Delta x_i} \left(\phi_{i+1/2}^n - \phi_{i-1/2}^n \right) \tag{3.6}$$

where

$$\phi_{i+1/2}^n \simeq \frac{1}{\Delta t} \int_{t^n}^{t^{n+1}} f(\boldsymbol{W}(x_{i+1/2}, t)) dt. \tag{3.7}$$

Equation (3.6) is the finite volume scheme for the $1D$ conservation laws, (3.1).

Clearly, it is in conservation form, (3.3) with the numerical flux, ϕ defined in (3.7) being the approximation of the physical flux, f across the cell interface at $x_{i+1/2}$. The main difficulty in the finite volume scheme is in computing the numerical fluxes, $\phi^n_{i\pm1/2}$. At each cell interface, the approximate solution, $\boldsymbol{W}^n_i$ is discontinuous, so it has to be estimated from the cell averages of the adjacent cells. The procedure to approximate these interface fluxes leads to different finite volume schemes. Godunov-type finite volume methods use the wave propagation information of the PDEs to construct the numerical fluxes. They utilize the solution of Riemann problems defined at cell interfaces. These methods and few centred methods are briefly described in appendix A. In the following sections, we present the finite volume schemes for the 2D flood model and the 1D channel model both in the presence of friction.

3.4 Finite Volume Method for the Flood Model

In this section, we present the finite volume scheme for the flood model - the 2D shallow water equations presented in chapter 2.3. We first consider the system without any source term, next we add bottom topography and finally friction terms are considered.

3.4.1 The 2D Shallow Water Equations without Source Terms

Here, we consider the system without source term. Recall the 2D shallow water equations 2.32 without source term, namely

$$\begin{aligned}
&\partial_t \Pi + \nabla \cdot F(\Pi) = 0, \\
&\text{where } \Pi = \begin{pmatrix} H \\ q_x \\ q_y \end{pmatrix}, \quad F(\Pi) = (F_1(\Pi), F_2(\Pi)), \\
&F_1(\Pi) = \begin{pmatrix} q_x \\ \frac{q_x^2}{H} + \frac{1}{2}gH^2 \\ \frac{q_x q_y}{H} \end{pmatrix}, F_2(\Pi) = \begin{pmatrix} q_y \\ \frac{q_x q_y}{H} \\ \frac{q_y^2}{H} + \frac{1}{2}gH^2 \end{pmatrix}.
\end{aligned} \tag{3.8}$$

Recall that this model, (3.8) satisfies the rotation invariance property [Toro, 2001], namely

$$F(\Pi) \cdot \vec{n} = T_n^{-1} F_1(T_n \Pi), \tag{3.9}$$

for all Π and for all vectors, $\vec{n} = (n_x, n_y)^T$, where T_n is a rotation matrix which depends on the vector, $\vec{n}$ and is given by

$$T_n = \begin{pmatrix} 1 & 0 & 0 \\ 0 & n_x & n_y \\ 0 & -n_y & n_x \end{pmatrix}, \quad T_n^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & n_x & -n_y \\ 0 & n_y & n_x \end{pmatrix} \tag{3.10}$$

and T_n^{-1} is the inverse matrix of T_n [Toro, 2001]. We now derive the finite volume scheme for the model.

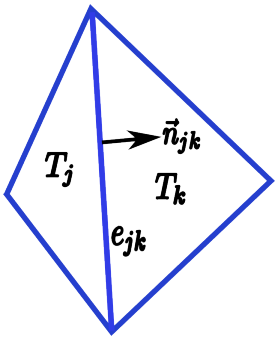


Figure 3.2: 2D mesh showing two neighbour cells, T_j and T_k, the edge e_{jk} between them and the normal vector $\vec{n}_{jk}$.

Let Ω_h be a mesh of the physical domain, $\Omega \in \mathbb{R}^2$ consisting of polygonal subsets of Ω. Let $T_j \in \Omega_h$ be an element of Ω_h , regarded as a control volume or cell, see figure 3.2 Also, let $T_k \in \Omega_h$ be a neighbour cell to T_j and e_{jk} is the edge between T_j and T_k, while $\vec{n}_{jk}$ is a unit vector normal to edge e_{jk} and outward to T_j. Furthermore, let $|T_j|$ and $|e_{jk}|$ be the area of T_j and length of e_{jk} respectively and let $\mathcal{E}_j$ be the set of all edges of T_j. Integrating (3.8) over T_j, one gets

$$\int_{T_j} \partial_t \Pi d\vec{x} + \int_{T_j} \nabla \cdot F(\Pi) d\vec{x} = 0.$$

Use of Gauss divergence theorem leads to

$$\int_{T_j} \partial_t \Pi d\vec{x} + \int_{\partial T_j} F(\Pi) \cdot \vec{n}_j d\omega_j = 0, \tag{3.11}$$

where ∂T_j is the boundary of T_j, $\vec{n}_j$ the outward unit normal to ∂T_j and $d\omega_j$ is a boundary element of ∂T_j. As pointed out in section 3.3, we aim to approximate cell

averages of the true solution, so we define the following average.

$$\Pi_j = \frac{1}{|T_j|} \int_{T_j} \Pi \, d\vec{x}. \tag{3.12}$$

Since T_j is polygonal in shape, the boundary is made of straight lines, hence the second integral in (3.11) can be written as.

$$\int_{\partial T_j} F(\Pi) \cdot \vec{n}_j d\omega_j = \sum_{e_{jk} \in \mathcal{E}_j} \int_{e_{jk}} F(\Pi) \cdot \vec{n}_{jk} de_{jk}, \tag{3.13}$$

where $de_{jk} = d\omega_j|_{e_{jk}}$. Hence (3.11) becomes

$$|T_j| \frac{d\Pi_j}{dt} = - \sum_{e_{jk} \in \mathcal{E}_j} \int_{e_{jk}} F(\Pi) \cdot \vec{n}_{jk} \, de_{jk}. \tag{3.14}$$

By the rotation invariance property, (3.9), we have

$$|T_j| \frac{d\Pi_j}{dt} = - \sum_{e_{jk} \in \mathcal{E}_j} \int_{e_{jk}} T^{-1}_{\vec{n}_{jk}} F_1(T_{\vec{n}_{jk}} \Pi) \, de_{jk}. \tag{3.15}$$

Note that $F_1(T_{\vec{n}_{jk}} \Pi)\Big|_{e_{kj}}$ is the evaluation of $F_1(\Pi)$ across the edge, e_{jk} using $T_{\vec{n}_{jk}} \Pi|_{e_{jk}}$. However, $T_{\vec{n}_{jk}} \Pi = T_{\vec{n}_{jk}} \Pi|_{e_{jk}}$ is discontinuous on the edge since $\Pi|_{e_{jk}}$ (Π on the edge) is discontinuous. So, we use the 1D numerical flux function, ϕ alongside the two cell averages, Π_j, Π_k to approximate, $F_1(T_{\vec{n}_{jk}} \Pi)\Big|_{e_{kj}}$. That is,

$$F_1(T_{\vec{n}_{jk}} \Pi)\Big|_{e_{kj}} \approx \phi(T_{\vec{n}_{jk}} \Pi_j, T_{\vec{n}_{jk}} \Pi_k), \tag{3.16}$$

where ϕ is a 1D numerical flux function consistent with F_1, (see appendix A) hence (3.15) becomes

$$\begin{aligned} |T_j| \frac{d\Pi_j}{dt} &= - \sum_{e_{jk} \in \mathcal{E}_j} \int_{e_{jk}} T^{-1}_{\vec{n}_{jk}} \phi(T_{\vec{n}_{jk}} \Pi_j, T_{\vec{n}_{jk}} \Pi_k) \, de_{jk} \\ &= - \sum_{e_{jk} \in \mathcal{E}_j} |e_{jk}| T^{-1}_{\vec{n}_{jk}} \phi(T_{\vec{n}_{jk}} \Pi_j, T_{\vec{n}_{jk}} \Pi_k). \end{aligned} \tag{3.17}$$

In this thesis, we use the HLL solver [Harten et al., 1983] discussed in appendix A.

Using the first-order forward Euler time discretization, we have the first-order

explicit finite volume scheme for the 2D shallow water equation, (3.8) as

$$\Pi_j^{n+1} = \Pi_j^n - \frac{\Delta t}{|T_j|} \sum_{e_{jk} \in \mathcal{E}_j} |e_{jk}| T_{\vec{n}_{jk}}^{-1} \phi(T_{\vec{n}_{jk}} \Pi_j^n, T_{\vec{n}_{jk}} \Pi_k^n). \tag{3.18}$$

Note that the quantity, $T_{\vec{n}_{jk}}^{-1} \phi(T_{\vec{n}_{jk}} \Pi_j^n, T_{\vec{n}_{jk}} \Pi_k^n)$ is the average 2D flux across the edge, e_{jk}. So, we denote this 2D flux function by ϕ^{2D}, that is,

$$\phi^{2D}(\Pi_j^n, \Pi_k^n, \vec{n}_{jk}) = T_{\vec{n}_{jk}}^{-1} \phi(T_{\vec{n}_{jk}} \Pi_j^n, T_{\vec{n}_{jk}} \Pi_k^n). \tag{3.19}$$

Then, the 2D scheme, (3.18) is same as

$$\Pi_j^{n+1} = \Pi_j^n - \frac{\Delta t}{|T_j|} \sum_{e_{jk} \in \mathcal{E}_j} |e_{jk}| \phi^{2D}(\Pi_j^n, \Pi_k^n, \vec{n}_{jk}). \tag{3.20}$$

where ϕ^{2D} is defined in (3.19).

3.4.2 The 2D Shallow Water Equations with Bottom Topography

We are now in position to discuss the numerical scheme for the flood model in the presence of bottom topography, (2.32), namely

$$\begin{aligned} &\partial_t \Pi + \nabla . F(\Pi) = S(\Pi, z_b), \\ &\Pi = \begin{pmatrix} H \\ q_x \\ q_y \end{pmatrix}, S(\Pi, z_b) = \begin{pmatrix} 0 \\ -gH\partial_x z_b(\vec{\boldsymbol{X}}) \\ -gH\partial_y z_b(\vec{\boldsymbol{X}}), \end{pmatrix}, \end{aligned} \tag{3.21}$$

where F is the same as in (3.8). The model, (3.21) presents more numerical challenges than the previous model, (3.8) due to the presence of the source term. The challenge is to be able to design numerical schemes that maintain steady state solutions. A steady state solution satisfies the following equation:

$$\nabla \cdot F(\Pi) = S(\Pi, z_b). \tag{3.22}$$

Hence we require schemes that would balance these terms, ($\nabla \cdot F(\Pi)$ and $S(\Pi, z_b)$) in the case of steady state conditions. Greenberg and Leroux [1996] introduced the concept of well-balanced schemes, and referred to them as schemes that satisfy steady state solutions at discrete level. In this thesis, we are particularly interested in steady state of water at rest. The C-property is introduced in [Bermudez and Vazquez, 1994]. A scheme is said to satisfy the C-property if the scheme satisfies

the steady state of water at rest [Bermudez and Vazquez, 1994]. Throughout this thesis, we will still use the term, well-balance, even though we actually refer to well-balance of steady state at rest. In this thesis, we use the well balanced scheme of [Audusse et al., 2004], see also [Audusse and Bristeau, 2005]. This scheme is called the hydrostatic reconstruction method (HRM). A review of well-balanced methods is presented in [Wang, 2011].

Consider the 2D mesh, Ω_h described in the previous section, with all the symbols retaining their meaning. Centred discretization of the numerical source term, $S_j^{2D,n} \approx \frac{1}{|T_j|}\int_{T_j} S(\Pi, z)d\vec{x}$ does not generally verify the steady state solutions [Katsaounis et al., 2004], so interface discretization of both the source and fluxes is an effective way to satisfy steady state solution [Bouchut, 2007; Katsaounis et al., 2004]. Hence, a general form to represent well-balanced schemes for (3.21) is the upwind interface form, namely

$$\Pi_j^{n+1} = \Pi_j^n - \frac{\Delta t}{|T_j|}\sum_{e_{jk}\in\mathcal{E}_j} |e_{jk}|\Big(\phi_s^{2D}(\Pi_j^n, \Pi_k^n, z_{b,j}, z_{b,k}, \vec{n}_{jk}) + S^{2D}(\Pi_j^n, \Pi_k^n, z_{b,j}, z_{b,k}, \vec{n}_{jk})\Big), \tag{3.23}$$

where ϕ_s^{2D} is the source-dependent interface numerical flux function and S^{2D} is an interface numerical source function. Each well-balanced method has its own definitions for the functions, ϕ_s^{2D} and S^{2D}.

As stated above, we are interested in the HRM for which these two functions are defined in the following. Consider the interface (edge), e_{jk} described in the mesh Ω_h above. Given the cell average vector, Π_j and bottom elevation, $z_{b,j}$ in cell, j and those - Π_k and $z_{b,k}$ of the neighbour cell, k. Then, the following steps are followed:

$$\tilde{H}_p^n = \max(H_p^n + z_{b,p} - \max(z_{b,j}, z_{b,k})), \quad p = j, k. \tag{3.24}$$

Compute : $T_{\vec{n}_{jk}}\Pi_j^n$ and $T_{\vec{n}_{jk}}\Pi_k^n$, then $T_{\vec{n}_{jk}}^{-1}\Pi_j^n$ and $T_{\vec{n}_{jk}}^{-1}\Pi_k^n$.

$$\widetilde{T_{\vec{n}_{jk}}\Pi_p^n} = \frac{\tilde{H}_p^n}{H_p^n} T_{\vec{n}_{jk}}\Pi_p^n, \quad p = j, k \tag{3.25}$$

$$\phi_s^{2D}(\Pi_j^n, \Pi_k^n, z_{b,j}, z_{b,k}, \vec{n}_{jk}) = T_{\vec{n}_{jk}}^{-1}\phi(\widetilde{T_{\vec{n}_{jk}}\Pi_j^n}, \widetilde{T_{\vec{n}_{jk}}\Pi_k^n}). \tag{3.26}$$

$$S^{2D}(\Pi_j^n, \Pi_k^n, z_{b,j}, z_{b,k}, \vec{n}_{jk}) = T_{\vec{n}_{jk}}^{-1}S^{hrm}(H_j^n, \tilde{H}_j^n). \tag{3.27}$$

where

$$S^{hrm}(a, \tilde{a}) = \begin{pmatrix} 0 \\ \frac{g}{2}(a^2 - \tilde{a}^2) \\ 0 \end{pmatrix} \quad \text{for any } a, \tilde{a} \in \mathbb{R}^+, \tag{3.28}$$

and ϕ is a 1D numerical flux function consistent with F_1. Hence, the first-order HRM in 2D is summarised as follows:

$$\Pi_j^{n+1} = \Pi_j^n - \frac{\Delta t}{|T_j|}\sum_{e_{jk}\in\mathcal{E}_j}|e_{jk}|\left(T_{\vec{n}_{jk}}^{-1}\phi(\widetilde{T_{\vec{n}_{jk}}\Pi_j^n}, \widetilde{T_{\vec{n}_{jk}}\Pi_k^n}) + T_{\vec{n}_{jk}}^{-1}S^{hrm}(H_j^n, \tilde{H}_j^n)\right). \tag{3.29}$$

The HRM, (3.28) has been widely used for solving the shallow water equations and shallow water type models, see for example [Audusse and Bristeau, 2005; Bouchut et al., 2004; Bouchut, 2007; Mangeney-Castelnau et al., 2005; Noelle et al., 2006; Mungkasi, 2012]. The properties of the scheme, such as being well balanced, mass conservative, positivity preserving, are established [Audusse et al., 2004; Bouchut, 2004, 2007]. It's simplicity and robustness is the reason why it is widely used, hence we adopt it in this thesis.

3.4.3 The 2D Shallow Water Equations with Bottom Topography and Friction

Finally, we consider the numerical schemes for the flood model in the presence of bottom friction, namely

$$\begin{aligned} &\partial_t\Pi + \nabla\cdot F(\Pi) = S(\Pi, z_b) + S_b(\Pi) \\ &S_b(\Pi) = \begin{pmatrix} 0 \\ -g\frac{n^2}{H^{7/3}}\vec{\boldsymbol{q}}|\vec{\boldsymbol{q}}| \end{pmatrix} \end{aligned} \tag{3.30}$$

where n is manning coefficient, S_b is the friction term and every other term is as defined in (3.21). The finite volume scheme for the model, (3.30) is the addition of a centred discritization of the friction term to the scheme (equation (3.29)) of the

model without friction. That is

$$\Pi_j^{n+1} = \Pi_j^n - \frac{\Delta t}{|T_j|} \sum_{e_{jk} \in \mathcal{E}_j} |e_{jk}| \left(T_{\vec{n}_{jk}}^{-1} \phi(\widetilde{T_{\vec{n}_{jk}} \Pi_j^n}, \widetilde{T_{\vec{n}_{jk}} \Pi_k^n}) + T_{\vec{n}_{jk}}^{-1} S^{hrm}(H_j^n, \tilde{H}_j^n) \right) + \Delta t S_b(\Pi_j^n). \tag{3.31}$$

In section 3.8, we present a few numerical experiments to verify that we correctly implemented this scheme.

3.5 Finite Volume Method for the Channel Flow Model

In this section, one known finite volume scheme for the numerical solution of the St Venant Model, (2.67), in presence of friction, is briefly presented. The model with friction is the following.

$$\begin{aligned} &\partial_t A + \partial_x Q = 0. \\ &\partial_t Q + \partial_x \left(\frac{Q^2}{A} \right) = -gA\partial_x \bar{\eta} + gAS_f. \end{aligned} \tag{3.32}$$

where $A, Q, \bar{\eta}$ are all defined in (2.67), S_f is the channel friction slope. It depends on the rate of flow, Q and given by

$$S_f = \frac{Q|Q|}{K^2}. \tag{3.33}$$

K is called the conveyance [Cunge et al., 1980; MacDonald, 1996] and defined as

$$K = \frac{A^{k_1}}{nP^{k_2}}. \tag{3.34}$$

P is the wetted perimeter of channel cross-section, $k_1 = 5/3, k_2 = 2/3$ and n is the Manning coefficient.

For the numerical solution of the St Venant model above, we consider the same 1D grid as described in section 3.1. So to each cell, K_i, we associate the following cell average discrete quantities $\boldsymbol{W}_i^n = (A_i^n, Q_i^n)^T$ at time level, t_n. Here, we outline the channel solver due to [Morales-Hernández et al., 2012] as summarised in [Morales-Hernández et al., 2013]. This scheme is widely used for the SVM, see for example [Morales-Hernández, 2014; Morales-Hernández et al., 2016]. It is well balanced, mass conservative and positivity preserving [Burguete and García-Navarro, 2001; Morales-Hernández, 2014].

The scheme is based on the formulation of the St Venant model as presented in [Cunge et al., 1980] and rewrites the model in the quasi-linear form.

$$\partial_t \boldsymbol{W} + J(\boldsymbol{W}, B)\partial_x \boldsymbol{W} = s'(x, \boldsymbol{W}). \tag{3.35}$$

where $\boldsymbol{W} = (A, Q)^T$, the Jacobian matrix,$J$ is given by

$$J(\boldsymbol{W}, B) = \begin{pmatrix} 0 & 1 \\ c^2 - \underline{u}^2 & 2\underline{u}. \end{pmatrix}, \underline{u} = \frac{Q}{A}, c = \sqrt{g\frac{A}{B}}, \tag{3.36}$$

$$s'(x, \boldsymbol{W}) = \begin{pmatrix} 0 \\ gA\left[S_o - S_f - \frac{d\underline{H}}{dx} + \frac{1}{B}\frac{dA}{dx}\right] \end{pmatrix}. \tag{3.37}$$

B is the top width at the free-surface, $\underline{H} = \mathcal{H}(A, Z_b(x))$ is the water depth from the 1D bottom elevation, $Z_b(x)$ to the flat free-surface, $\bar{\eta}$ and $S_o = -\frac{dZ_b}{dx}$ is the negative of channel bed slope. Details about this formulation can be found in [Morales-Hernández et al., 2013]. The eigenvalues and eigenvectors of $J(w, B)$ are found, namely

$$\lambda_1(\boldsymbol{W}, B) = \underline{u} - c, \quad \lambda_2(w, B) = \underline{u} + c$$

and

$$\mathbf{e}_1(\boldsymbol{W}, B) = (1, \lambda_1(\boldsymbol{W}, B))^T, \quad \mathbf{e}_2(\boldsymbol{W}, B) = (1, \lambda_2(\boldsymbol{W}, B))^T$$

respectively.

The numerical scheme (see [Morales-Hernández et al., 2013, 2012]) is then given by

$$\boldsymbol{W}_i^{n+1} = \boldsymbol{W}_i^n - \frac{\Delta t}{\Delta x}\left[\sum_{m=1}^{2}\left(\hat{\gamma}_m^+ \hat{\mathbf{e}}_m\right)_{i-1/2} + \sum_{m=1}^{2}\left(\hat{\gamma}_m^- \hat{\mathbf{e}}_m\right)_{i+1/2}\right]^n \tag{3.38}$$

where

$$\left(\hat{\gamma}_m^{\pm}\right)_{i+1/2} = \left[\frac{1}{2}[1 \pm sgn(\hat{\lambda})]\hat{\gamma} \pm \hat{\nu}\hat{\alpha}\right]_{m,i+1/2}, \quad m = 1, 2 \tag{3.39}$$

$$\left(\hat{\gamma}_m\right)_{i+1/2} = \left(\hat{\lambda}\hat{\alpha} - \hat{\beta}\right)_{m,i+1/2}, \quad m = 1, 2 \tag{3.40}$$

$\hat{\nu}$ is an artificial viscosity (entropy fix) to correct the entropy problem associated

with the Roe method [Morales-Hernández et al., 2013] and is given by

$$(\hat{\nu}_m)_{i+1/2} = \begin{cases} \frac{1}{4}\Big[(\lambda_m)_{i+1} - (\lambda_m)_i\Big], & \text{if } (\lambda_m)_i < 0 < (\lambda_m)_{i+1} \\ 0, & \text{else} \end{cases} \tag{3.41}$$

$m = 1, 2$

The Roe averaged eigenvalues and eigenvectors are

$$(\hat{\lambda}_m)_{i+1/2} := \lambda_m(\hat{\boldsymbol{W}}_{i+2}, \hat{B}_{i+1/2}), \quad (\hat{\mathbf{e}}_m)_{i+1/2} := \mathbf{e}_m(\hat{\boldsymbol{W}}_{i+1/2}, \hat{B}_{i+1/2}), \quad m = 1, 2. \tag{3.42}$$

And the Roe average vector, $\hat{\boldsymbol{W}}_{i+1/2} = \begin{pmatrix} \hat{A}_{i+1/2} \\ \hat{A}_{i+1/2}\hat{\underline{u}}_{i+1/2} \end{pmatrix}$ is given by

$$\hat{\underline{u}}_{i+1/2} = \frac{\sqrt{A_i}\underline{u}_i + \sqrt{A_{i+1}}\underline{u}_{i+1}}{\sqrt{A_i} + \sqrt{A_{i+1}}}, \tag{3.43}$$

$$\hat{A}_{i+1/2} = \frac{1}{2}(A_i + A_{i+1}). \tag{3.44}$$

Also

$$\hat{B}_{i+1/2} = \frac{1}{2}(B_i + B_{i+1}), \tag{3.45}$$

$$\hat{\underline{H}}_{i+1/2} = \left(\frac{\hat{A}}{\hat{B}}\right)_{i+1/2}, \quad \hat{c}_{i+1/2} = \sqrt{g\hat{\underline{H}}_{i+1/2}}. \tag{3.46}$$

Note that these Roe averages satisfy

$$\hat{A} = \hat{\underline{H}}\hat{B},$$

which is a requirement for the well-balanced property of the method. The wave strengths are given by

$$(\hat{\alpha}_1)_{i+1/2} = \left[\frac{\hat{\lambda}_2\Delta A - \Delta Q}{2\hat{c}}\right]_{i+/2}, \quad (\hat{\alpha}_2)_{i+1/2} = \left[\frac{-\hat{\lambda}_1\Delta A + \Delta Q}{2\hat{c}}\right]_{i+1/2}. \tag{3.47}$$

Finally,

$$(\hat{\beta}_1)_{i+1/2} = \left(-g\frac{\hat{A}}{2\hat{c}}\left[(\hat{S}_0 - \hat{S}_f)\Delta x - \Delta\underline{H} + \frac{1}{\hat{B}}\Delta A\right]\right)_{i+1/2}, \tag{3.48}$$

$$(\hat{\beta}_2)_{i+1/2} = -(\hat{\beta}_1)_{i+1/2}, \tag{3.49}$$

$$(\hat{S_o})_{i+1/2} = \frac{Z_{b,i+1} - Z_{b,i}}{x_{i+1} - x_i}, \quad (\hat{S_f})_{i+1/2} = S_f(\hat{w}_{i+1/2}). \tag{3.50}$$

where $(\Delta p)_{i+1/2} = p_{i+1} - p_i$ for any quantity, p. In section 3.8, we present numerical experiments to verify that we correctly implemented this scheme.

3.6 Boundary Conditions

Recall that to compute the interface numerical fluxes and/or sources, we require the cell average of the current cell and those of its neighbours. However, if the current cell is on the boundary of the domain, then at least one of its neighbours would not exist, so we would not have the needed cell averages to compute the interface fluxes and sources. These neighbour cell averages are obtained from the boundary conditions. There are several types of boundary conditions [Leveque, 2002; Toro, 1999; Roberts, 2013] but only two types are discussed here. They are solid and open boundaries. We briefly discuss them below. Let j denote the cell inside the computational domain and k, the fictitious boundary cell. Let $\vec{n} := (n_1, n_2)^T$, and $\vec{n}^{\perp} := (-n_2, n_1)^T$ be the outward normal and tangential vectors to the boundary. Define the following:

$$\vec{\boldsymbol{u}} = (\bar{u}, \bar{v})^T, \quad \text{the velocity vector.}$$

$$u_n := n_1\bar{u} + n_2\bar{v}, \quad \text{the velocity normal to the boundary.}$$

$$u_{n^{\perp}} := -n_2\bar{u} + n_1\bar{v}, \quad \text{the velocity tangential to the boundary.}$$

$$\hat{\Pi}_j = \begin{pmatrix} H_j \\ H_j u_{n,j} \\ H_j u_{n^{\perp},j} \end{pmatrix}, \quad \text{the known flow vector in cell, j.}$$

$$\hat{\Pi}_k = \begin{pmatrix} H_k \\ H_k u_{n,k} \\ H_k u_{n^{\perp},k} \end{pmatrix}, \quad \text{the } \textbf{unknown} \text{ flow vector in cell, k.}$$

where $u_{n,j}$ and $u_{n^{\perp},j}$ are the normal and tangential velocities in cell, j and $u_{n,k}$ and $u_{n^{\perp},k}$ are those of cell k. The goal here is to explain how to obtain $\hat{\Pi}_k$.

3.6.1 Solid Boundary

This type of boundary corresponds to a solid wall and requires that the velocity normal to it be zero [Bristeau and Coussin, 2001; Leveque, 2002], namely

$$\vec{\boldsymbol{u}} \cdot \vec{n} = 0. \tag{3.51}$$

For this boundary, we assume the continuity of the tangential velocity, $u_{n^\perp}$ and the water depth, H but reflect the normal velocity, u_n, hence we define

$$\hat{\Pi}_k = \begin{pmatrix} H_j \\ -H_j u_{n,j} \\ H_j u_{n^\perp,j} \end{pmatrix}. \tag{3.52}$$

[Bristeau and Coussin, 2001].

3.6.2 Open Boundary

There is no physical obstacle to flow on this type of boundary. The value to be prescribed in this type of boundary depends on the flow conditions. We consider the following 1D shallow water system projected in the normal direction, namely

$$\partial_t \hat{\Pi} + \partial_n F(\hat{\Pi}) \cdot \vec{n} = 0 \tag{3.53}$$

where $\hat{\Pi} = (H, Hu_n, Hu_{n^\perp})^T$ and F is the flux function in the 2D SWE. We consider only the following two of the eigenvalues of the Jacobian matrix of this system, $\lambda_1 = u_n - c$ and $\lambda_2 = u_n + c$ and their associated Riemann invariants, $r_1 = u_n - 2c, r_2 = u_n + 2c$, where $c = \sqrt{gH}$.

Define the Froud number,

$$F_r = \frac{|\vec{\boldsymbol{u}}|}{c}. \tag{3.54}$$

Then, the boundary values are specified depending on if the flow is sub-critical;

$$F_r < 1, \quad (|\vec{\boldsymbol{u}}| < c), \tag{3.55}$$

or super-critical;

$$F_r > 1, \quad (|\vec{\boldsymbol{u}}| > c). \tag{3.56}$$

Sub-critical Flow: $|\vec{\boldsymbol{u}}| < c$.

If $|\vec{\boldsymbol{u}}| < c$, then the eigenvalue, $\lambda_2 = u_n + c$ is positive. This means that λ_2 corresponds to an outgoing characteristics and the associated Riemann invariant must be constant along this characteristics, namely

$$u_{n,k} + 2c_k = u_{n,j} + 2c_j, \tag{3.57}$$

where $c_p = \sqrt{gH_p}$, $p = j, k$.

Then, (3.57) allows to compute $u_{n,k}$ if H is known on the boundary or to compute H_k if the velocity is imposed on the boundary [Hou et al., 2013, 2015; Bristeau and Coussin, 2001]. The tangential velocity, $u_{n^\perp,k}$ is assumed equal to that of the inner cell, j. Then, the x and y velocity components are recovered from

$$\bar{u} = u_n n_1 - u_{n^\perp} n_2, \tag{3.58}$$

$$\bar{v} = u_n n_2 + u_{n^\perp} n_1, \tag{3.59}$$

[Hou et al., 2013].

Super-critical Flow: $|\vec{\boldsymbol{u}}| > c$.

For an inflow boundary, the flow condition are all imposed and none is taken from the inner cell. This gives $\hat{\Pi}_k$. For the outflow boundaries, all the flow variables in the fictitious cell are taken from the inner cell, hence $\hat{\Pi}_k = \hat{\Pi}_j$.

Remark 3.6.1. *In all the experiments performed in this thesis, closed boundaries are treated exactly as explained in this section. But for the open boundaries whose boundary conditions are not explicitly imposed, we implement the transmissive boundary condition, that is we take all the flow variables from the inner cell, namely $\hat{\Pi}_k = \hat{\Pi}_j$. We adopt this approach since it is also widely used, see [Toro, 2001; Leveque, 2002; Toro, 1999].*

3.7 Time step calculations

Finally, we discuss how the time step sizes, Δt is chosen in our simulations. Time step sizes, Δt are computed based on the CFL condition, named after Courant, Friedrichs and Lewy [Leveque, 2002]. The CFL condition requires that the numerical domain of dependence contains the true domain of dependence of the PDE [Leveque, 1992, 2002] and this results to the requirement that numerical wave speeds do not travel a distance more than the size of one grid cell in one time step [Toro,

1999, 2001]. Here, we adopt the following for time step size calculations. For 1D simulations, we use

$$\Delta t = CFL \min_i \left(\frac{\Delta x_i}{\max_j \{|\lambda_j|\}_i} \right) \tag{3.60}$$

where CFL is called the CFL number or Courant number [Leveque, 2002; Toro, 1999, 2001] and it satisfies

$$0 < CFL < 1$$

For 2D simulations, we use

$$\Delta t = CFL \min_{i,j} \left(\frac{\Delta x_{i,j}}{\max_k \{|\lambda_k|_{i,j}\}}, \frac{\Delta y_{i,j}}{\max_k \{|\lambda_k|\}_{i,j}} \right) \tag{3.61}$$

where λ_k is the maximum eigenvalue of Jacobians of F_1 and F_2. Other choices exists for 2D unstructured grids, see [Morales-Hernández et al., 2013; Hou et al., 2015] for example. In all our simulations we use $CFL = 0.95$.

3.8 Numerical Results

In this section, we briefly present a few test cases to verify that we correctly implemented the numerical schemes described in the previous sections of this chapter. The detailed numerical analysis of the schemes are out of the scope of this thesis. We also present a 2D simulation of flow in a channel connected to a floodplain and use the results to show that it is indeed beneficial to couple 1D and 2D simulations as claimed in the literature and in this thesis. We remark that both the 1D and 2D solvers have been tested for several tests cases from literature including the tests for well balance of lake at rest and mass conservation. Here, we only report a few test cases for each solver to demonstrate that they are correctly implemented to achieve their intended purposes.

3.8.1 Test Cases for the 1D Channel Flow Solver

We present two test cases for the 1D scheme.

1D test case 1 : Riemann problem in a constant width channel

The first test consists of a Riemann problem in a constant width, flat bottom and 50.0 metre long channel with an initial discontinuity located at 10 metres from the

beginning of the channel. The initial condition is the following.

$$\underline{H}(x,0) = \begin{cases} 1.0, & \text{if } x < 10.0 \\ 0.1, & \text{if } x \geq 10.0 \end{cases}, \qquad \underline{u}(x,0) = \begin{cases} 2.5, & \text{if } x < 10.0 \\ 0.0, & \text{if } x \geq 10.0 \end{cases}$$

For this problem, we used open boundaries conditions on both ends of the domain. This test case is from chapter 7 in [Toro, 2001] where the exact solutions are also presented. This exact solution is also implement for the purpose of this thesis and used to compare the result of the scheme being discussed here. The simulation was run for 7.0 seconds using a grid of 500 cells.

The results of the exact and numerical solutions for the free-surface elevation and velocity after four different time steps are plotted in figure 3.3 . The plots are for the solutions at the 100th, 200th, 400th and 465th time steps which, respectively, correspond to the following times 1.5 seconds, 3.01357 seconds, 6.02277 seconds and 7 seconds of the simulation. It can seen that the numerical solutions for both free surface and velocity reasonably approximate the exact solutions at all the outputted time steps.

As a further validation of the scheme, we consider the experimental order of convergence (EOC). Let w be an exact solution which we want to approximate and w_h be a numerical approximation obtained using a grid size of $h = \Delta x$. If there is a constant C independent of h, such that

$$|w_h - w| \leq Ch^p, \tag{3.62}$$

for sufficiently small h, then the method is said to be of order p. We consider the l_1-norm error of our solution, namely

$$Error = \frac{1}{N} \sum_{i=1}^{N} |w_i - \bar{w}(x_i)| \tag{3.63}$$

where, N is the number of grid cells, w_i and $\bar{w}$, the approximate and exact cell average of the solution in the cell with center at x_i. Then the EOC is computed on a sequence of simulations with decreasing grid sizes

$$h\left(\frac{1}{2}\right)^n, \quad \text{for } n = 0, 1, 2, ... \tag{3.64}$$

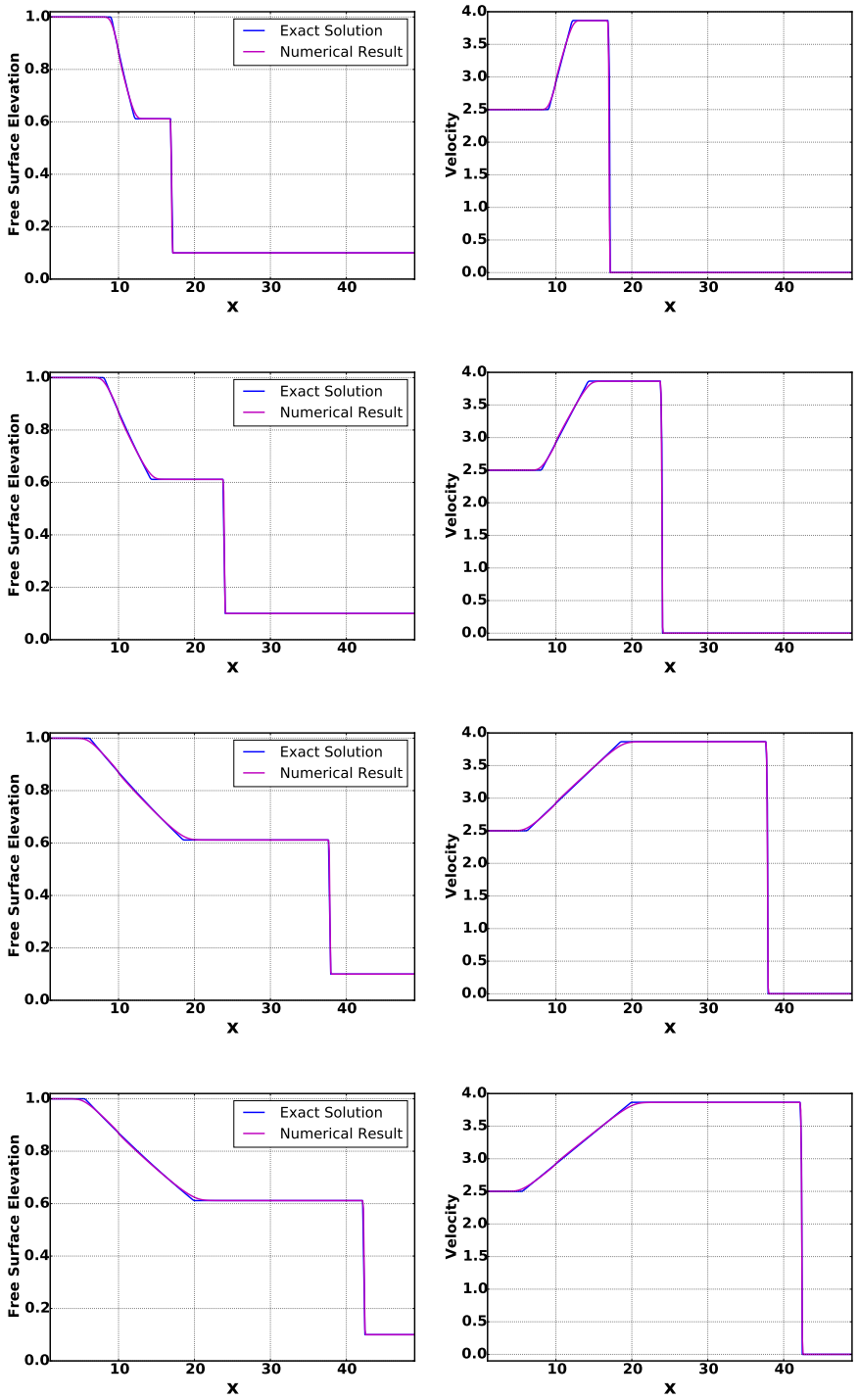


Figure 3.3: Exact and Numerical Results for the Riemann problem showing the free surface elevation (left column) and velocity (right column) at different time steps. From the top to the bottom are the results at the 100th, 200th, 400th and the last (465th) time steps. In each plot, the exact and numerical solutions are in blue and magenta respectively.

where h is the coarsest (initial) grid size. Specifically, EOC is defined as

$$p = \frac{log\left(\frac{E_h}{E_{h/2}}\right)}{log2} \tag{3.65}$$

where E_h and $E_{h/2}$ are the errors for approximate solutions obtained with grids sizes h and $h/2$ respectively.

Table 3.1 shows the errors and experimental order of convergence for both wetted cross sectional area and discharge. As stated before, all the methods discussed in this thesis are first-order methods. It is clear that the order of convergence of the method for both quantities tend to one. The difficulty in reaching one, exactly, is due to the fact that the problem contains discontinuity.

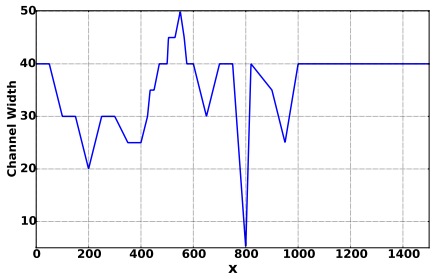
N	E_A	EOC_A	E_Q	EOC_Q
10	0.147049	-	0.377779	-
20	0.0753928	0.963797	0.189512	0.99525
40	0.0412884	0.868692	0.0958896	0.982844
80	0.0212781	0.956365	0.0504271	0.927175
160	0.0114398	0.89531	0.0263537	0.936192
320	0.00610385	0.906268	0.0134899	0.966129
640	0.0032658	0.902286	0.00712752	0.920407

Table 3.1: Experimental Order of Convergence for the 1D Riemann Problem (1D test case 1). E_A, E_Q are the errors in the cross sectional area, A and discharge, Q respectively, while EOC_A and EOC_Q are their respective orders of accuracy.

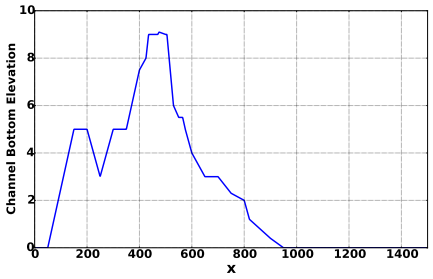
1D test 2 : 1D Flow over a channel with varying geometry

In the next test case we simulate the flow in a 1500 meter long locally rectangular channel with varying width and non-flat bottom as used in [Morales-Hernández et al., 2012]. The bottom and width of the channel are shown in figure 3.4 and their measurements are obtained from [Vázquez-Cendón, 1999]. The initial conditions are zero velocity and constant free surface elevation of 12 metres. The boundary conditions are

$$\begin{aligned} &Q(1500, t) = 0.0 \\ &H(0, t) = 16.0 + 4sin\left(\frac{(t - 10800)\pi}{21600}\right) \end{aligned} \tag{3.66}$$



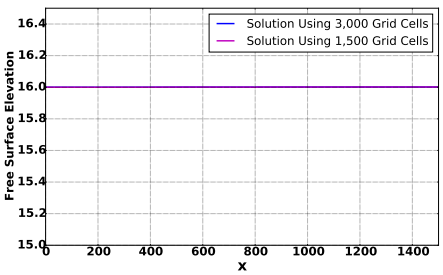
(a) The channel width variation for 1D test case 2.



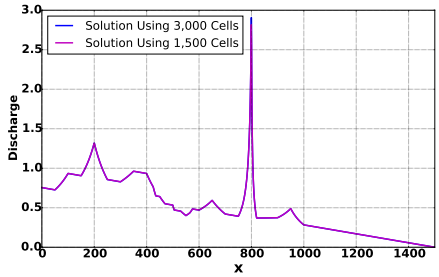
(b) The channel bottom elevation for the 1D test case 2.

Figure 3.4: The channel width and bed variations for the 1D flow in varying geometry channel.

The numerical solution is computed with a grid of 1,500 cells and the results after 10,800 seconds are shown in figure 3.5. To validation our implementation, we used a reference solution obtained on a refined mesh with 3,000 cells. The solutions are shown in figure 3.5. It can be see that both solutions are are very close. Again, the solution obtained here agrees with those in the literature, particularly, those presented in [Morales-Hernández et al., 2012]. This proves that our implementation of the 1D solver is correct.



(a) Free-surface elevation, $\bar{\eta} = \underline{H} + Z_b$ after 10,800 seconds.



(b) The discharge, $q = \underline{H}u(m^2/s)$ after 10,800 seconds.

Figure 3.5: Numerical Solution for the flow in channel with varying geometry (1D test 2).

3.8.2 Test Cases For the 2D Floodplain Flow Solver

We now turn to the 2D scheme.

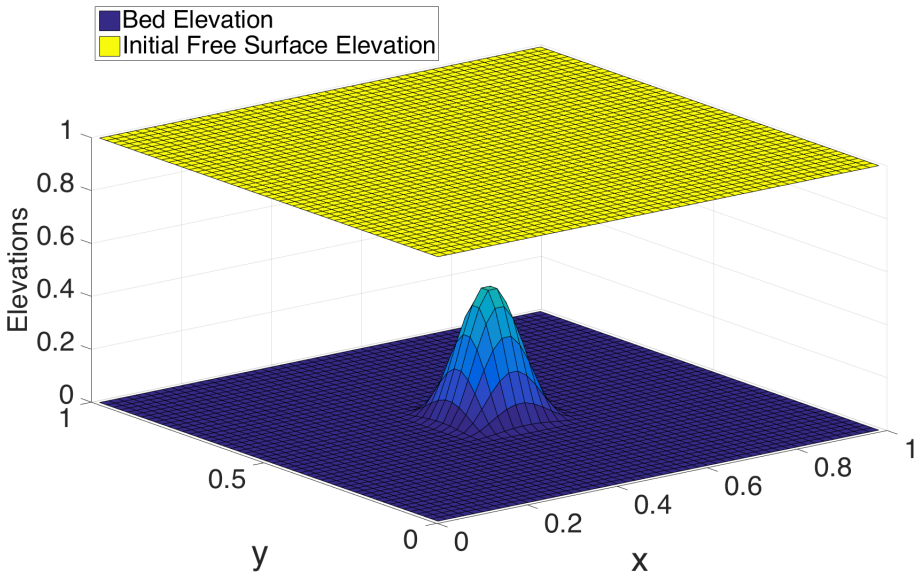


Figure 3.6: Bed elevation and initial free surface elevation for still water

2D test case 1: Still water over complex bottom topography

The test case for 2D solver is to check its ability to preserve well balance of lake at rest. Consider a 2D domain, $[0,1] \times [0,1]$ whose bottom elevation is given by

$$z_b(x,y) = \begin{cases} z_0(x,y), & \text{if } 0.375 \leq x \leq 0.625 \text{ and } 0.375 \leq y \leq 0.625 \\ 0, & \text{otherwise.} \end{cases} \tag{3.67}$$

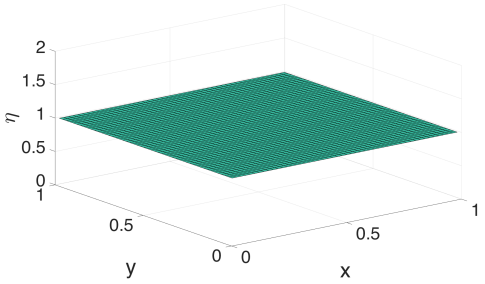
where

$$z_0(x,y) = 0.5\left[\cos\left(\frac{(x-0.5)\pi}{0.25}\right)\cos\left(\frac{(y-0.5)\pi}{0.25}\right)\right]^2 \tag{3.68}$$

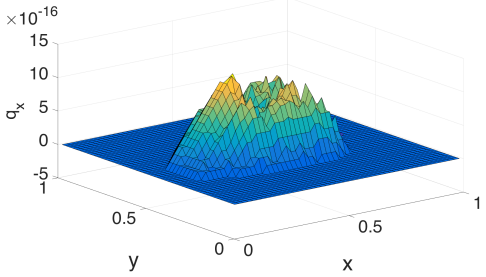
The initial condition is given as follows

$$\begin{aligned} H(x,y,0) &= 1 - z_b(x,y) \\ \bar{u}(x,y,0) &= \bar{v}(x,y,0) = 0 \end{aligned} \tag{3.69}$$

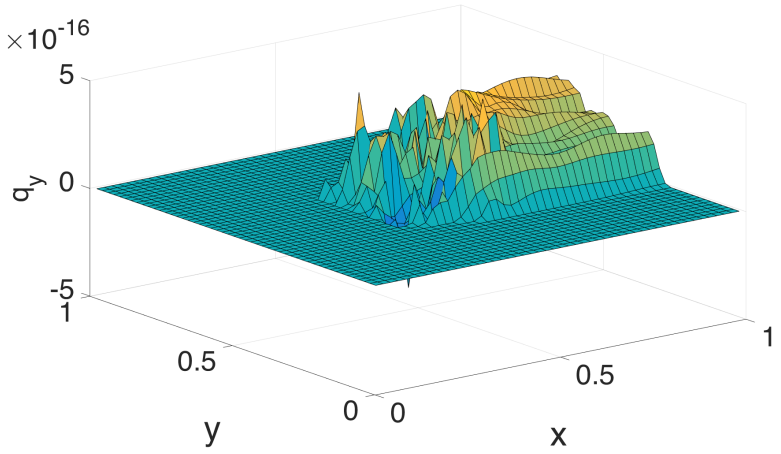
This problem corresponds to a still water over a complex bottom topography, see figure 3.6. The numerical solution is computed with a 50-by-50 cell grid and the results after 5 seconds are shown in figure 3.7. As depicted, the steady state of water at rest is preserved in the sense that the free surface elevation remains unchanged while discharges are correct with very small errors of the order of 10^{-16}.



(a) Free-surface Elevation



(b) Discharge along x-direction



(c) Discharge along y-direction

Figure 3.7: Numerical Results after 5 seconds for the still water over complex bottom topography

2D Test 2 : Oscillatory Flow in a Parabolic Basin

The problem of the 2D flow over a parabolic basin, whose analytical solution (in presence of Coriolis force) is derived in [Thacker, 1981], is next considered. The bottom topography is given by

$$z_b(x,y) = h_0 + z_0(x,y), \tag{3.70}$$

where

$$z_0(x,y) = h_0\left(\frac{(x-\frac{L}{2})^2 + (y-\frac{L}{2})^2}{a^2} - 1\right), \quad \forall (x,y) \in [0,L]\times[0,L], \tag{3.71}$$

where h_0, a and L are constants. Like in [Delestre et al., 2013], the Coriolis force is ignored in this thesis, so the Thacker's analytical solution takes the following form:

$$\begin{aligned} &H(x,y,t) = 2\frac{\eta_0 h_0}{a^2}\left(\left(x-\frac{L}{2}\right)\cos\omega t + \left(x-\frac{L}{2}\right)\sin\omega t - \frac{\eta_0}{2}\right) - z_0(x,y). \\ &\bar{u}(x,y,t) = -\eta_0\omega\sin\omega t, \quad \bar{v}(x,y,t) = \eta_0\omega\cos\omega t. \end{aligned} \tag{3.72}$$

η_0 is a constant and $\omega = \sqrt{2\frac{gh_0}{a^2}}$. This corresponds to flow in which the water surface remains planar and rotates in the basin. As prescribed in [Delestre et al., 2013], we consider the following data: $a = 1, h_0 = 0.1, \eta_0 = 0.5$ and the domain length in each direction is $L = 4$. The initial condition is taken as the analytical solution at $t = 0$ (see figure 3.8). We consider the closed boundary conditions on all the walls.

We simulate this problem using a grid of 300 cells in each direction and the results outputted after 5.93018 seconds. Figure 3.9 displays the numerical (left column) and analytical (right column) solutions for the water height and the discharges. It can be seen that the numerical results reasonably approximate the analytical results even as this problem consists of wet/dry interfaces which presents much difficulty to numerical schemes.

As a further validation, this test case is used to compute the rate of convergence for the 2D scheme in l_1-norm. For this, we run the simulate up to $t = 1.0$ with initial grid of size 10×10. Table 3.2 shows the errors, in l_1-norm, for the conserved variables alongside their convergence rates. Convergence rates for all variables are seen to be approaching one which as expected for the scheme being examined. This verifies that the scheme is properly implemented.

With the above results, we are now satisfied that the schemes presented in the previous sections are correctly implemented to achieve the intended purposes.

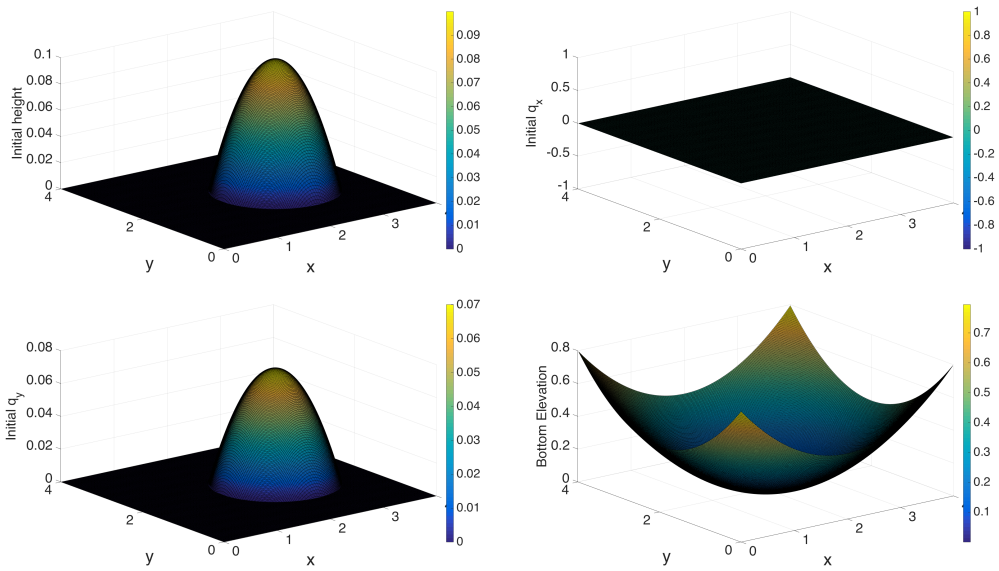


Figure 3.8: The bottom elevation and the initial water depth and discharges for 2D test case 2.

Steps	Grid	E_H	EOC_H	E_{q_x}	EOC_{q_x}	E_{q_y}	EOC_{q_y}
0.4	10×10	0.0601382	-	0.0511435	-	0.0501766	-
0.2	20×20	0.0338552	0.828904	0.0284981	0.843682	0.0291977	0.78116
0.1	40×40	0.020067	0.754555	0.0174113	0.710842	0.0163561	0.836022
0.05	80×80	0.0115893	0.792035	0.0101739	0.775146	0.00884623	0.886699
0.025	160×160	0.00654927	0.82338	0.00577636	0.816646	0.00470902	0.909636
0.0125	320×320	0.00362899	0.851769	0.00321985	0.843169	0.00247111	0.930266

Table 3.2: Experimental Order of convergence for the 2D scheme using the flow in parabolic basin problem (2D test 2). E_H, E_{q_x} and E_{q_y} are the errors in l_1-norm of H, q_x and q_y respectively, while EOC_H, EOC_{q_x} and EOC_{q_y} are their respective experimental order of convergence.

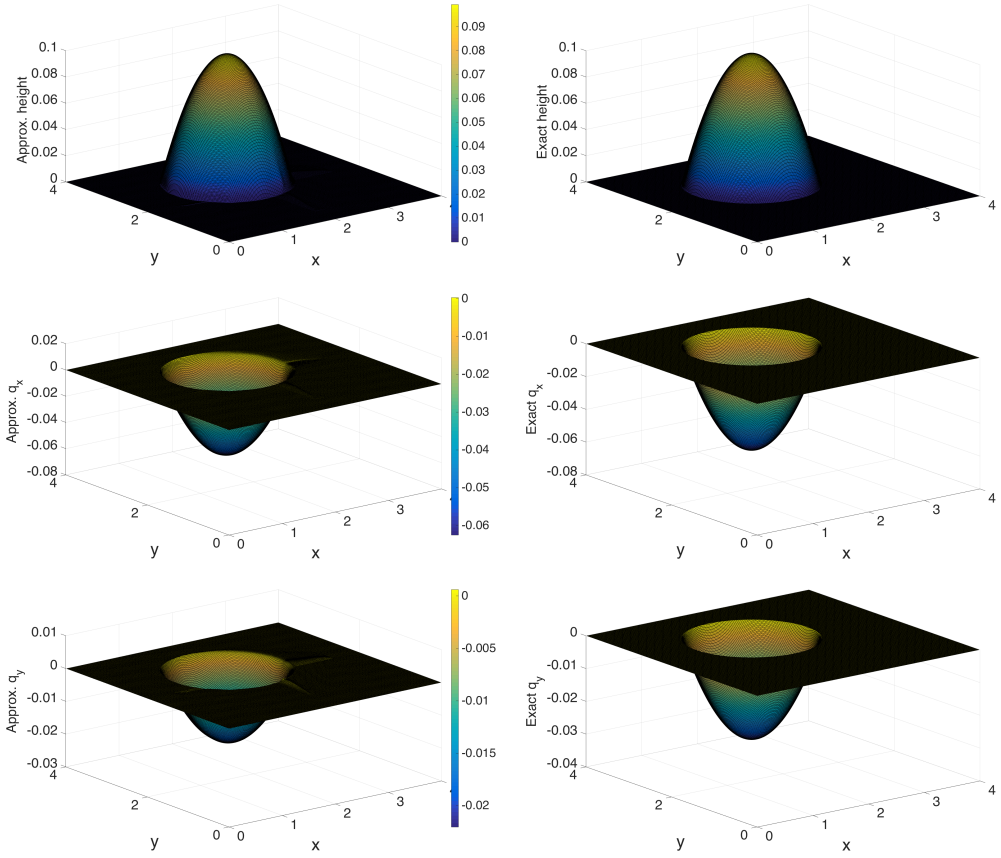


Figure 3.9: Numerical (left) and analytical (right) results for the water height and the discharges after 5.93018 seconds for 2D test case 2.

3.8.3 Why Coupling is needed

Finally, we demonstrate why combining 2D and 1D solvers is necessary for efficient simulation of river flooding problems. For this, we use the problem considered in [Morales-Hernández et al., 2013]. This problem consists of a dam break flow in a 19.3 meter long, 0.5 meter constant width flat channel which ends into a flat floodplain, see figure 3.10. The National Laboratory of Civil Engineering in the IST in Portugal designed and measured this test case [Viseu et al., 1999; Morales-Hernández et al., 2013]. A reservoir is located from the left end of the channel to 6.10 metres away (position of dam in figure 3.10). The initial condition is

$$H(x,y,0) = \begin{cases} 0.504, & \text{at the reservoir, that is } 0 \le x \le 6.10 \text{ and } 1.8 \le y \le 2.3 \\ 0.003, & \text{elsewhere} \end{cases}$$

$$\bar{u}(x,y,0) = \bar{v}(x,y,0) = 0 \quad \text{everywhere}$$

The manning coefficient for both channel and floodplain is $0.009\text{s/m}^{1/3}$ and the boundaries are all closed walls except the right side as indicated in figure 3.10. The labels $P_1, P_2, \ldots, P_6$ are probe points in the flow domain. More about this test case can be found in [Morales-Hernández et al., 2013; Viseu et al., 1999].

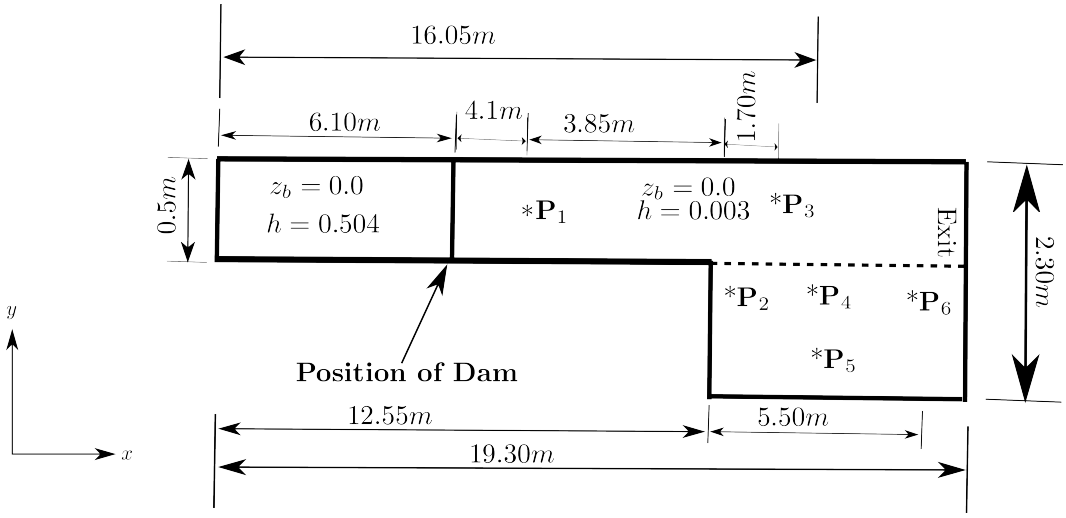


Figure 3.10: Top view of Channel and Floodplain for river-flooding problem

The flow is simulated with a 193×91 grid. Figure 3.11 displays the free surface elevation after ten seconds while figure 3.12 compares the results of the simulation with those of experiment at the probe points, $P_1, P_2, \ldots, P_6$. It can be

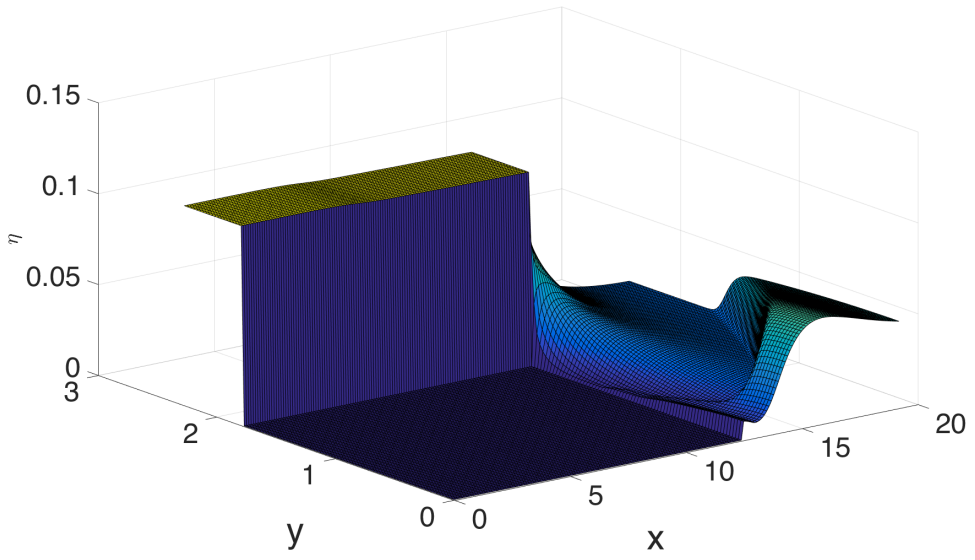


Figure 3.11: Final Free Surface Elevation, η after ten seconds.

seen that the numerical results reasonably approximate the experimental results, especially the wave arrival times at the various probe points.

Figures 3.13 display scatter plots of flow quantities (η, q_x and q_y) within the river channel (that is, points in $[0, 19.3] \times [1.8, 2.3] \in \mathbb{R}^2$ in figure 3.10). We observe the following. (i) From figure 3.13(a), we see that the free surface elevation is a single value at any point in the channel ($0 \leq x \leq 12.55$ for all $y \in [1.8, 2.3]$) before the channel/floodplain part($x > 12.55$). This means that the free surface does not vary laterally (in y) within the channel if flooding does not occur. On the other hand, for points ($x > 12.55$) where channel connects with floodplain, the free surface elevation vary laterally (in y). However, these variations are small. (ii) The observation for η in (i) above is also observed for q_x and q_y in figures 3.13(b) and 3.13(c) respectively. That is, q_x and q_y do not vary laterally when not overflowing. (iii) From figure 3.13(c), the $y-$discharge, q_y remains at absolute zero if not flooding($x \leq 12.55$) but varies when flooding ($x > 12.55$).

From the above observations, we conclude the following (a) the flow is absolutely 1D in the channel if not flooding and (b) even in the event of flooding, the lateral variations are small. However, recall that this simulation uses 91 cells, that grid spacing of 0.0252 along y-direction even inside the channel. This means that a huge number of 20 (approximately) cells in $y-$direction were used within the channel even when these cells all have the same solution because the flow is 1D.

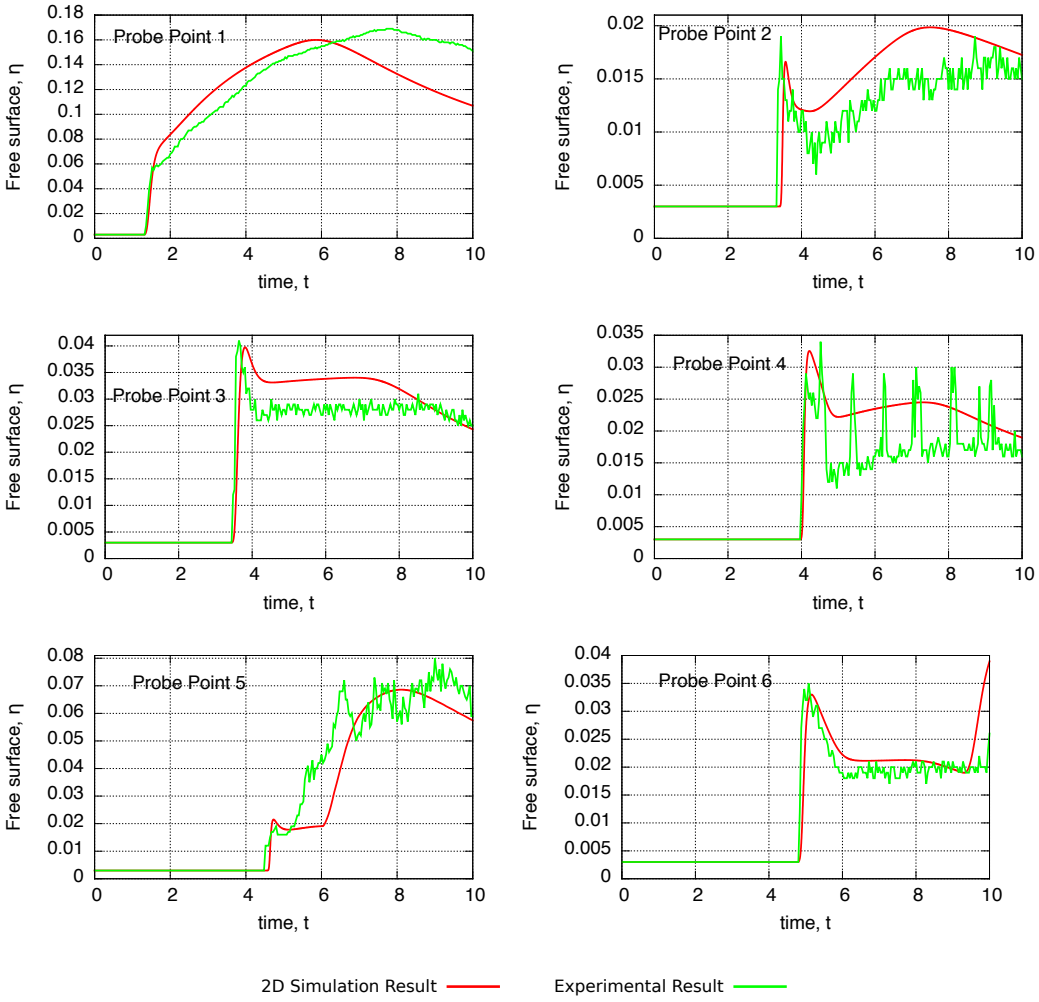
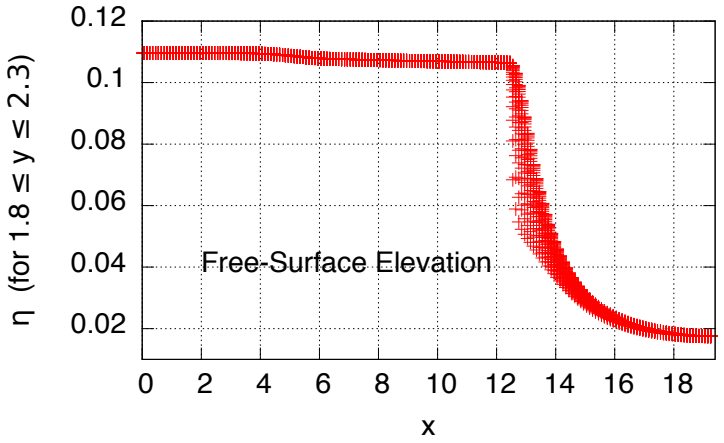


Figure 3.12: Comparison of the simulation results with the experimental data for the free surface elevation, η after the last time step. The positions of the probe points are indicated in figure 3.10.

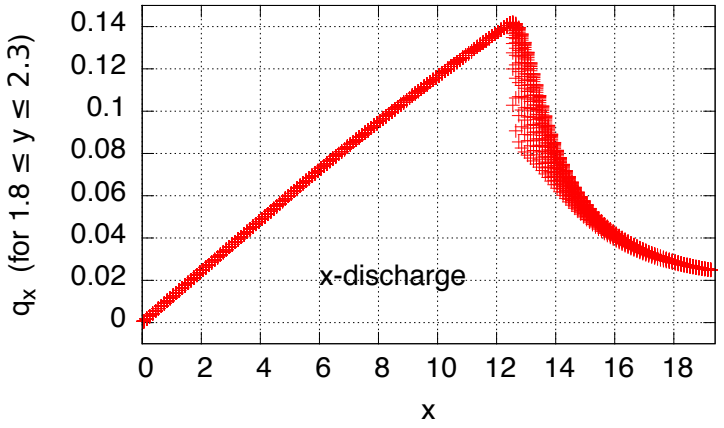
This is an avoidable computational expensive simulation.

Therefore, it is expedient to apply 1D simulation within the channel and retain 2D simulations in the floodplains. This has been the subject of much research and also the focus of the subsequent chapters of this thesis. Precisely, this thesis emphasis on two points which have not been given too much attention in the literature, namely (i) As revealed in figure 3.13(c), there is lateral variations in q_y (also in η and q_x) during flooding, hence the first coupling method proposed in this thesis aims to restore the variations in q_y while retaining 1D simulations within the channel. (ii) As revealed in figures 3.13(a)-3.13(c), all flow quantities actually vary laterally (even though small) during flooding. Therefore, the second method proposed in this thesis is a family of methods that aim to restore all flow variations whenever flooding. The goal of the methods proposed in this thesis is to reproduce the full 2D simulation results at reduced computational cost, not to reproduce the experimental results. We therefore remark the following.

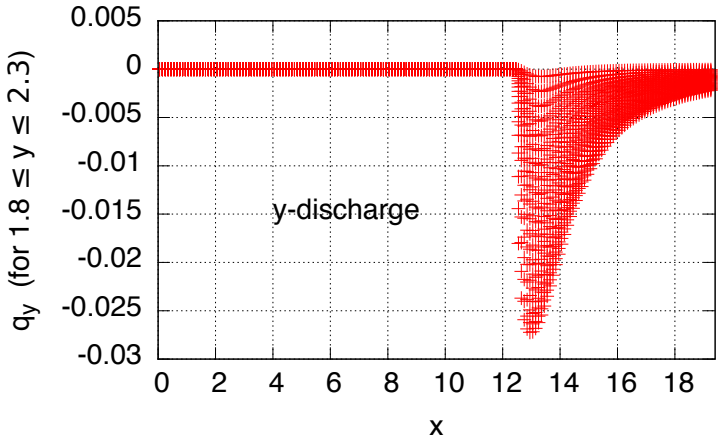
Remark 3.8.1. *Throughout this thesis, the full 2D simulation results, like those obtained this section, are considered as the benchmark results which we wish our coupled methods in subsequent chapters to reproduce, at best. We do not aim to reproduce the experimental results.*



(a) Lateral variation of free surface, η in the channel $(1.8 \leq y \leq 2.3)$



(b) Lateral variation of x-discharge, q_x in the channel, $(1.8 \leq y \leq 2.3)$



(c) Lateral variation of y-discharge, q_y in the channel, $(1.8 \leq y \leq 2.3)$

Figure 3.13: Lateral Variation of the flow quantities within the channel after ten seconds (the last time step).

Chapter 4

The Horizontal Coupling Method

4.1 Introduction

In this chapter, we propose a coupling method following the lines of existing methods. The proposed method adopts the derivation of the Saint Venant Models presented in section 2.4. By taking into account a completely filled channel, we reuse the more general form of the models derived in section 2.4 to formulate the proposed method. We classify the proposed method as a horizontal coupling method since we still assume that the free-surface elevation is laterally horizontal in the channel. Hence, throughout this thesis, we use the name "the horizontal coupling method (HCM)" to identify this method. The derivation of the models follows the lines of [Marin and Monnier, 2009], however, we do not impose or use any restriction on the channel width variation in order to derive the resulting coupling term. The coupling term so obtained is intuitively simple to compute in the context of conservative numerical methods.

A fundamental issue we consider here is the channel lateral discharge during overflow. By physical intuition, during overflow like flooding or draining, water flows out of or into the channel from both of its lateral boundaries. This means that lateral velocities (or discharges) at both sides are in opposite directions and very likely to differ in magnitude. Therefore, the lateral discharge will rarely be constant across the channel cross sections, even though the free-surface elevation remains constant over the cross section. Consequently, we propose that different discharges for each lateral boundary of a given cross section, need to be computed. We employ the 2D vertical-component shallow water equation (2.31), as an ad-hoc

model to compute these lateral discharges.

The numerical approximation for the channel model with coupling terms, is proposed in the context of finite volume methods. The numerical scheme is shown to be globally well balanced with respect to lake at rest. The "no numerical flooding" property is introduced and the proposed scheme is equipped to preserve this property through the use of hydrostatic reconstruction [Audusse et al., 2004]. This property is also proved to hold.

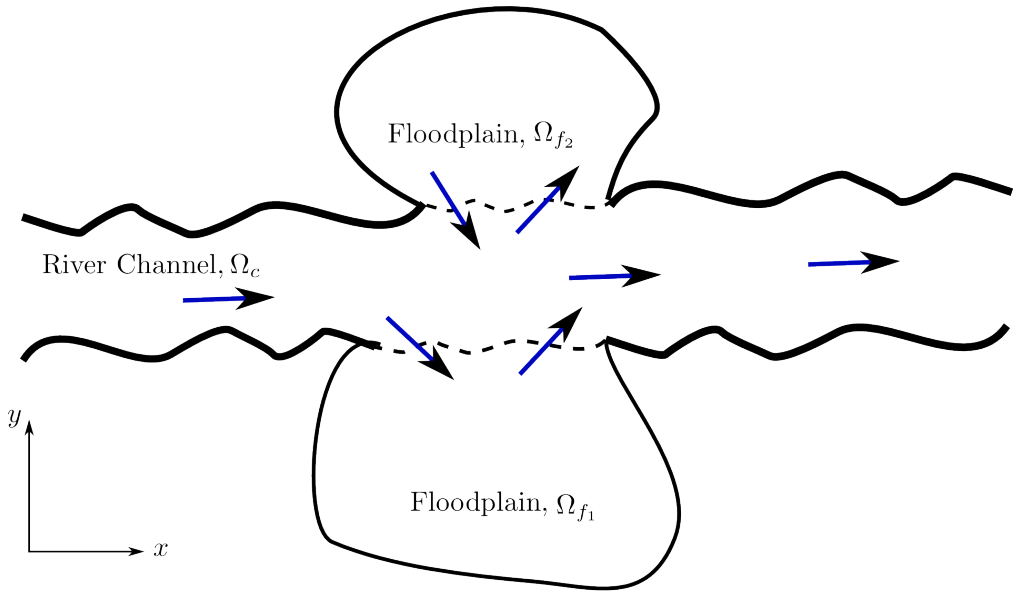
For channels whose walls are parallel to the x-axis, a finite volume approximation of our resulting model coincides with the flux based method (FBM) proposed in [Bladé et al., 2012], if we do not compute the lateral discharges as proposed here. However, for more general cases, the finite volume approximation of our model, differs from the FBM of [Bladé et al., 2012].

The remaining of this chapter is organised as follows. In section 4.2, we present the 1D channel models with coupling term and also write down the ad-hoc model for the lateral discharges. The 2D models for flood simulations are stated in section 4.3. A general numerical scheme for the 1D models and a detailed presentation of the discrete coupling term are presented in section 4.5. We also present the numerical scheme for the lateral discharge model in this section. Then the properties of the method are discussed in section 4.6 where we introduce the concept of "no-numerical flooding property" and show that the method preserves this property and is also well-balanced. In section 4.7, we suggest one implementation strategy for the proposed coupling method. Finally, in section 4.8, we present one numerical test to access the performance of the method. More experiments are presented in chapter 6.

4.2 The Channel Model with Coupling Terms

The task at hand is to couple the flow of water in the sub domains, Ω_c and Ω_f, depicted in figures 4.1 (top view) and 4.2 (cross sectional view). As we stated in chapter 1.3, the flow in the floodplain regions are described by the 2D shallow water equations (2.32), so we focus on the flow in the channel, Ω_c throughout this section. To do this, we decouple the channel from the rest of the domain, see figure 4.3. Then, we focus on deriving the model specifically for the flow in the channel region.

The coupled model we seek to present follows exactly the assumptions of section 2.4 for flow in the channel, Ω_c except that we allow for the possibility of a full channel cross section. Figure 4.3 depicts this situation. All channel symbols retain their meanings from figure 2.2. Since the cross section is possibly full, we



where $\Omega_f = \Omega_{f_1} \cup \Omega_{f_2}$,
$\Omega^2 = \Omega_f \cup \Omega_c$

Figure 4.1: Top view of 2D flow domain, $\Omega^2 = \Omega_c \cup \Omega_f$ consisting of the river channel, Ω_c and the floodplains, $\Omega_f = \Omega_{f_1} \cup \Omega_{f_2}$.

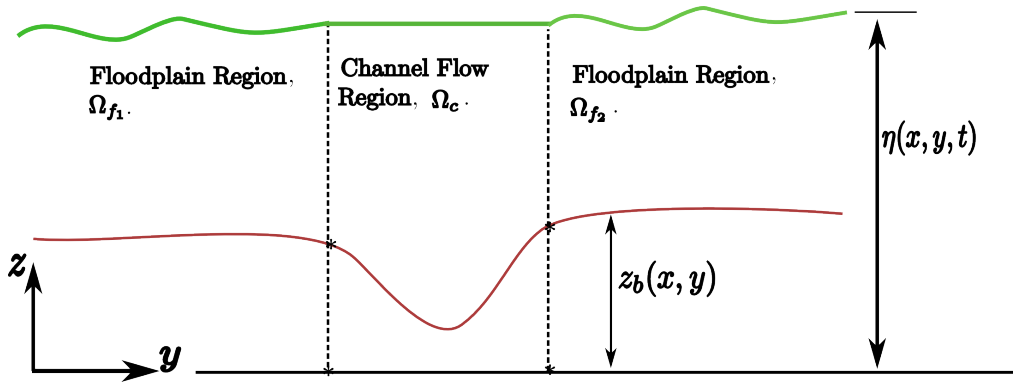


Figure 4.2: Cross sectional view of 2D flow domain, $\Omega^2 = \Omega_c \cup \Omega_f$ consisting of the river channel, Ω_c and the floodplains, $\Omega_f = \Omega_{f_1} \cup \Omega_{f_2}$.

extend the definition of cross section width functions, $B(x,z), y_{l,r}(x,z)$ to include the region above channel wall elevation (i.e. where $z \geq z_b^w(x)$) as follows:

$$\left.\begin{aligned} B(x,z) &= B(x, z_b^w(x)), \\ y_{l,r}(x,z) &= y_{l,r}^w(x) := y_{l,r}(x, z_b^w(x)), \end{aligned}\right\} \quad \forall \quad z \geq z_b^w(x). \tag{4.1}$$

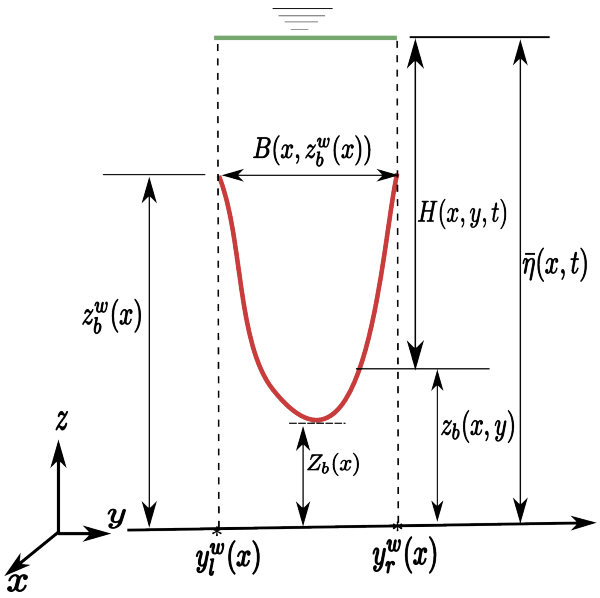


Figure 4.3: Channel cross section in HCM without the floodplains, showing the channel bottom topography, $z_b(x,y)$ (in red), the channel wall elevation , $z_b^w(x)$, laterally flat free-surface elevation $\bar{\eta}(x,t)$, the bottom elevation in 1D sense $Z_b(x)$, the top width $B(x, z_b^w(x))$ and the $y-$coordinates $y_l^w(x) := y_l(x, z_b^w(x))$ and $y_r^w(x) := y_r(x, z_b^w(x))$ respectively of the left and right lateral walls at the channel top. $H(x,y,t)$ is the depth of water measured from $z_b(x,y)$ to the flat free surface $\bar{\eta}(x,t)$.

So that

$$\left.\begin{aligned} &B(x,\bar{\eta}(x,t)) = B(x, z_b^w(x)),\\ &y_{l,r}(x,\bar{\eta}(x,t)) = y_{l,r}(x, z_b^w(x)), \end{aligned}\right\} \quad \text{whenever} \quad \bar{\eta}(x,t) \geq z_b^w(x). \tag{4.2}$$

Therefore,

$$z_b(\vec{\boldsymbol{X}})|_{y=y_{l,r}(x,\bar{\eta}(x,t))} = z_b(\vec{\boldsymbol{X}})|_{y=y_{l,r}^w(x)} = z_b^w(x) \text{ whenever } \bar{\eta}(x,t) \geq z_b^w(x). \tag{4.3}$$

In this last equation, the first equality follows from (4.2), while the second equality follows from equation (2.39). This allows to simplify the lower limits appearing in the coupling terms and also the lateral functions.

The fact that the channel might be full means that equations (2.40) and (2.41) no longer hold. Therefore, all the integrals on the right hand sides of equations (2.56) and (2.65) no longer vanish. That is, $\Phi^A(x,t)$ in (2.57) and $\Phi^Q(x,t)$ in (2.64) no

longer vanish where

$$\begin{aligned}\Phi^A(x,t) = \partial_x y_r(x,\bar{\eta}) \left[\int_{z_b(\vec{\boldsymbol{X}})}^{\bar{\eta}(x,t)} u dz\right]_{y=y_r(x,\bar{\eta})} - \left[\int_{z_b(\vec{\boldsymbol{X}})}^{\bar{\eta}(x,t)} v dz\right]_{y=y_r(x,\bar{\eta})} \\ - \partial_x y_l(x,\bar{\eta}) \left[\int_{z_b(\vec{\boldsymbol{X}})}^{\bar{\eta}(x,t)} u dz\right]_{y=y_l(x,\bar{\eta})} + \left[\int_{z_b(\vec{\boldsymbol{X}})}^{\bar{\eta}(x,t)} v dz\right]_{y=y_l(x,\bar{\eta})}.\end{aligned} \tag{4.4}$$

$$\begin{aligned}\Phi^Q(x,t) = \partial_x y_r(x,\bar{\eta}) \left[\int_{z_b(\vec{\boldsymbol{X}})}^{\bar{\eta}(x,t)} u^2 dz\right]_{y=y_r(x,\bar{\eta})} - \left[\int_{z_b(\vec{\boldsymbol{X}})}^{\bar{\eta}(x,t)} uv dz\right]_{y=y_r(x,\bar{\eta})} \\ - \partial_x y_l(x,\bar{\eta}) \left[\int_{z_b(\vec{\boldsymbol{X}})}^{\bar{\eta}(x,t)} u^2 dz\right]_{y=y_l(x,\bar{\eta})} + \left[\int_{z_b(\vec{\boldsymbol{X}})}^{\bar{\eta}(x,t)} uv dz\right]_{y=y_l(x,\bar{\eta})}.\end{aligned} \tag{4.5}$$

Hence, we obtain the 1D channel model with coupling term as equations (2.56) and (2.65), namely

$$\partial_t A(x,t) + \partial_x Q(x,t) = \Phi^A(x,t). \tag{4.6}$$

$$\partial_t Q(x,t) + \partial_x \left(\frac{Q^2(x,t)}{A(x,t)}\right) = -gA(x,t)\partial_x \bar{\eta}(x,t) + \Phi^Q(x,t). \tag{4.7}$$

We note that $\Phi^A(x,t)$ and $\Phi^Q(x,t)$ are non-zero only when channel is full, otherwise they are zero because the limits of integration would then be equal. Since the coupling term is zero except when, $\bar{\eta}(x,t) \geq z_b^w(x)$, that is when $y_{l,r}(x,\bar{\eta}(x,t)) = y_{l,r}(x, z_b^w(x))$, hence we can replace the lateral points in the coupling terms with $y_{l,r}(x, z_b^w(x))$ and denote them as

$$y_l^w(x) = y_l(x, z_b^w(x)) \text{ and } y_r^w(x) = y_r(x, z_b^w(x)). \tag{4.8}$$

To proceed, let us recall from (2.26) and (2.27) the following quantities of

2D SWEs.

$$
\begin{aligned}
q_x(\vec{\boldsymbol{X}},t) &= \int_{z_b(\vec{\boldsymbol{X}})}^{\eta(\vec{\boldsymbol{X}},t)} u(\vec{\boldsymbol{X}},z,t)dz,\\
q_y(\vec{\boldsymbol{X}},t) &= \int_{z_b(\vec{\boldsymbol{X}})}^{\eta(\vec{\boldsymbol{X}},t)} v(\vec{\boldsymbol{X}},z,t)dz,\\
\int_{z_b(\vec{\boldsymbol{X}})}^{\eta(\vec{\boldsymbol{X}},t)} u^2(\vec{\boldsymbol{X}},z,t)dz &\simeq \frac{q_x^2}{H}, \quad \int_{z_b(\vec{\boldsymbol{X}})}^{\eta(\vec{\boldsymbol{X}},t)} u(\vec{\boldsymbol{X}},z,t)v(\vec{\boldsymbol{X}},z,t)dz \simeq \frac{q_x q_y}{H}.
\end{aligned}
\tag{4.9}
$$

Using equations (4.9) the coupling terms become

$$
\Phi^A(x,t) = q_x|_{y=y_r^w(x)}\partial_x y_r^w(x) - q_y|_{y=y_r^w(x)} - q_x|_{y=y_l^w(x)}\partial_x y_l^w(x) + q_y|_{y=y_l^w(x)}.
\tag{4.10}
$$

$$
\Phi^Q(x,t) = \left.\frac{q_x^2}{H}\right|_{y=y_r^w(x)}\partial_x y_r^w(x) - \left.\frac{q_x q_y}{H}\right|_{y=y_r^w(x)} - \left.\frac{q_x^2}{H}\right|_{y=y_l^w(x)}\partial_x y_l^w(x) + \left.\frac{q_x q_y}{H}\right|_{y=y_l^w(x)}.
\tag{4.11}
$$

Therefore, the 1D channel model with coupling terms can be summarised as

$$
\partial_t A(x,t) + \partial_x Q(x,t) = \Phi_L^A(x,t) + \Phi_R^A(x,t).
\tag{4.12}
$$

$$
\partial_t Q(x,t) + \partial_x\left(\frac{Q^2(x,t)}{A(x,t)}\right) = -gA(x,t)\partial_x\bar{\eta}(x,t) + \Phi_L^Q(x,t) + \Phi_R^Q(x,t).
\tag{4.13}
$$

where

$$
\Phi_L^A(x,t) = \left(-q_x|_{y=y_l^w(x)}\partial_x y_l^w(x) + q_y|_{y=y_l^w(x)}\right).
\tag{4.14}
$$

$$
\Phi_R^A(x,t) = \left(q_x|_{y=y_r^w(x)}\partial_x y_r^w(x) - q_y|_{y=y_r^w(x)}\right).
\tag{4.15}
$$

$$
\Phi_L^Q(x,t) = \left(-\left.\frac{q_x^2}{H}\right|_{y=y_l^w(x)}\partial_x y_l^w(x) + \left.\frac{q_x q_y}{H}\right|_{y=y_l^w(x)}\right).
\tag{4.16}
$$

$$
\Phi_R^Q(x,t) = \left(\left.\frac{q_x^2}{H}\right|_{y=y_r^w(x)}\partial_x y_r^w(x) - \left.\frac{q_x q_y}{H}\right|_{y=y_r^w(x)}\right).
\tag{4.17}
$$

The difference between the approach adopted here and the one proposed in [Marin and Monnier, 2009] is in the following subsection.

4.2.1 Notational Simplification

The coupling terms above are not straight forward to identify or approximate but we can transform them into forms that can be easily recognised and computed, at least approximately. For this purpose, we reformulate them to functions of fluxes at channel lateral boundaries which are easily computed at least in the framework of conservative numerical methods, like the finite volume methods.

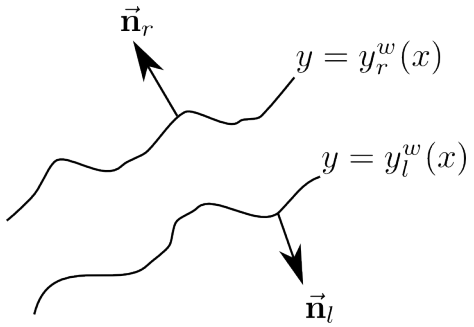


Figure 4.4: Top view of Lateral Boundaries (at elevation, $z = z_b^w(x)$)

Let $\vec{\boldsymbol{n}}_l = (n_l^x, n_l^y)^T$ and $\vec{\boldsymbol{n}}_r = (n_r^x, n_r^y)^T$ be the outward unit normal vectors to the lateral boundaries at $y = y_l^w(x)$ and $y = y_r^w(x)$ respectively (see figure 4.4). Then $m_l = \frac{n_l^y}{n_l^x}$ and $m_r = \frac{n_r^y}{n_r^x}$ are the slopes of the normal lines along $\vec{\boldsymbol{n}}_l, \vec{\boldsymbol{n}}_r$. Furthermore, by the theorem of perpendicular straight lines, the slopes $\partial_x y_l^w(x)$ and $\partial_x y_r^w(x)$ of tangents to $y_l^w(x)$ and $y_r^w(x)$ satisfy the following :

$$m_l \partial_x y_l^w(x) = -1, \quad m_r \partial_x y_r^w(x) = -1.$$

Therefore, we obtain the slopes of the lateral boundaries in terms of the outward unit normals as follows :

$$\partial_x y_l^w(x) = -\frac{n_l^x}{n_l^y}, \quad \partial_x y_r^w(x) = -\frac{n_r^x}{n_r^y}, \quad n_l^y, n_r^y \neq 0. \tag{4.18}$$

Define $\vec{\boldsymbol{q}} = (q_x, q_y)^T$, then using the above relations, we now rewrite the

coupling terms as follows :

$$\Phi_L^A(x,t) = \frac{n_l^x}{n_l^y} q_x(\vec{\boldsymbol{X}},t)|_{y=y_l^w(x)} + q_y(\vec{\boldsymbol{X}},t)|_{y=y_l^w(x)} = \frac{1}{n_l^y}\left(\vec{\boldsymbol{q}}\cdot\vec{\boldsymbol{n}}_l\right)\Big|_{y=y_l^w(x)}. \tag{4.19}$$

$$\Phi_R^A(x,t) = -\frac{n_r^x}{n_r^y} q_x(\vec{\boldsymbol{X}},t)|_{y=y_r^w(x)} - q_y(\vec{\boldsymbol{X}},t)|_{y=y_r^w(x)} = -\frac{1}{n_r^y}\left(\vec{\boldsymbol{q}}\cdot\vec{\boldsymbol{n}}_r\right)\Big|_{y=y_r^w(x)}. \tag{4.20}$$

$$\begin{aligned}
\Phi_L^Q(x,t) &= \frac{n_l^x}{n_l^y}\frac{q_x^2(\vec{\boldsymbol{X}},t)}{H(\vec{\boldsymbol{X}},t)}\Big|_{y=y_l^w(x)} + \frac{q_y(\vec{\boldsymbol{X}},t)q_y(\vec{\boldsymbol{X}},t)}{H(\vec{\boldsymbol{X}},t)}\Big|_{y=y_l^w(x)} \\
&= \frac{1}{n_l^y}\left(n_l^x\frac{q_x^2(\vec{\boldsymbol{X}},t)}{H(\vec{\boldsymbol{X}},t)} + n_l^y\frac{q_y(\vec{\boldsymbol{X}},t)q_y(\vec{\boldsymbol{X}},t)}{H(\vec{\boldsymbol{X}},t)}\right)\Big|_{y=y_l^w(x)} \\
&= \frac{1}{n_l^y}\left(n_l^x\left[\frac{q_x^2(\vec{\boldsymbol{X}},t)}{H(\vec{\boldsymbol{X}},t)} + \frac{g}{2}H^2(\vec{\boldsymbol{X}},t)\right] + n_l^y\frac{q_y(\vec{\boldsymbol{X}},t)q_y(\vec{\boldsymbol{X}},t)}{H(\vec{\boldsymbol{X}},t)}\right)\Big|_{y=y_l^w(x)} \\
&\quad - \frac{n_l^x}{n_l^y}\frac{g}{2}H^2(\vec{\boldsymbol{X}},t)\Big|_{y=y_l^w(x)}.
\end{aligned} \tag{4.21}$$

$$\begin{aligned}
\Phi_R^Q(x,t) &= -\frac{n_r^x}{n_r^y}\frac{q_x^2(\vec{\boldsymbol{X}},t)}{H(\vec{\boldsymbol{X}},t)}\Big|_{y=y_r^w(x)} - \frac{q_y(\vec{\boldsymbol{X}},t)q_y(\vec{\boldsymbol{X}},t)}{H(\vec{\boldsymbol{X}},t)}\Big|_{y=y_r^w(x)} \\
&= -\frac{1}{n_r^y}\left(n_r^x\left[\frac{q_x^2(\vec{\boldsymbol{X}},t)}{H(\vec{\boldsymbol{X}},t)} + \frac{g}{2}H^2(\vec{\boldsymbol{X}},t)\right] + n_r^y\frac{q_y(\vec{\boldsymbol{X}},t)q_y(\vec{\boldsymbol{X}},t)}{H(\vec{\boldsymbol{X}},t)}\right)\Big|_{y=y_r^w(x)} \\
&\quad + \frac{n_r^x}{n_r^y}\frac{g}{2}H^2(\vec{\boldsymbol{X}},t)\Big|_{y=y_r^w(x)}.
\end{aligned} \tag{4.22}$$

Let $f_L^1(x,t)$ and $f_L^2(x,t)$ be the first and second components respectively, of the analytical flux of the 2D SWE in direction of $\vec{\boldsymbol{n}}_l$, that is $F(\Pi(\vec{\boldsymbol{X}},t)).\vec{\boldsymbol{n}}_l\Big|_{y=y_l^w(x)}$. Then

$$f_L^1(x,t) = \left(\vec{\boldsymbol{q}}.\vec{\boldsymbol{n}}_l\right)\Big|_{y=y_l^w(x)} \text{ and} \tag{4.23}$$

$$f_L^2(x,t) = \left(n_l^x\left[\frac{q_x^2(\vec{\boldsymbol{X}},t)}{H(\vec{\boldsymbol{X}},t)} + \frac{g}{2}H^2(\vec{\boldsymbol{X}},t)\right] + n_l^y\frac{q_y(\vec{\boldsymbol{X}},t)q_y(\vec{\boldsymbol{X}},t)}{H(\vec{\boldsymbol{X}},t)}\right)\Big|_{y=y_l^w(x)}, \tag{4.24}$$

see the definition of $F(\Pi)$ in (3.8). Similarly, let the first and second components of

2D SWE flux in direction of $\vec{\boldsymbol{n}}_r$ be denoted $f_R^1(x,t)$ and $f_R^2(x,t)$ respectively, then

$$f_R^1(x,t) = \left(\vec{\boldsymbol{q}}.\vec{\boldsymbol{n}}_r\right)\Big|_{y=y_r^w(x)} \text{ and} \tag{4.25}$$

$$f_R^2(x,t) = \left(n_r^x\left[\frac{q_x^2(\vec{\boldsymbol{X}},t)}{H(\vec{\boldsymbol{X}},t)} + \frac{g}{2}H^2(\vec{\boldsymbol{X}},t)\right] + n_r^y\frac{q_y(\vec{\boldsymbol{X}},t)q_y(\vec{\boldsymbol{X}},t)}{H(\vec{\boldsymbol{X}},t)}\right)\Big|_{y=y_r^w(x)}. \tag{4.26}$$

Therefore, we get the coupling terms (4.19)-(4.22) in flux forms, namely

$$\Phi_L^A(x,t) = \frac{1}{n_l^y}f_L^1(x,t), \tag{4.27}$$

$$\Phi_R^A(x,t) = -\frac{1}{n_r^y}f_R^1(x,t), \tag{4.28}$$

$$\Phi_L^Q(x,t) = \frac{1}{n_l^y}f_L^2(x,t) - \frac{n_l^x}{n_l^y}\frac{g}{2}H^2(\vec{\boldsymbol{X}},t)\Big|_{y=y_l^w(x)}, \tag{4.29}$$

$$\Phi_R^Q(x,t) = -\frac{1}{n_r^y}f_R^2(x,t) + \frac{n_r^x}{n_r^y}\frac{g}{2}H^2(\vec{\boldsymbol{X}},t)\Big|_{y=y_r^w(x)}. \tag{4.30}$$

Hence, the 1D channel model with coupling terms is

$$\partial_t A(x,t) + \partial_x Q(x,t) = \frac{1}{n_l^y}f_L^1(x,t) - \frac{1}{n_r^y}f_R^1(x,t). \tag{4.31}$$

$$\begin{aligned}\partial_t Q(x,t) + \partial_x\left(\frac{Q^2(x,t)}{A(x,t)}\right) &= -gA(x,t)\partial_x\bar{\eta}(x,t) + \frac{1}{n_l^y}f_L^2(x,t)\\ &- \frac{n_l^x}{n_l^y}\frac{g}{2}H^2(\vec{\boldsymbol{X}},t)\Big|_{y=y_l^w(x)} - \frac{1}{n_r^y}f_R^2(x,t) + \frac{n_r^x}{n_r^y}\frac{g}{2}H^2(\vec{\boldsymbol{X}},t)\Big|_{y=y_r^w(x)}.\end{aligned} \tag{4.32}$$

where f_L^1 and f_L^2 are the first and second components respectively, of out-going 2D flux at the left boundary, $y = y_l^w(x)$, while f_R^1 and f_R^2 are the first and second components of the out-going 2D flux at the right lateral boundary $y = y_r^w(x)$.

The coupling term derived by [Marin and Monnier, 2009] which we denote by Φ^{mm} is the following

$$\Phi^{mm} = \begin{pmatrix} f_L^1(x,t) - f_R^1(x,t) \\ f_L^1(x,t)u|_{y=y_l^w(x)} - f_R^1(x,t)u|_{y=y_r^w(x)} \end{pmatrix} \tag{4.33}$$

Remark 4.2.1. *We obtained the above coupling terms without using or imposing any restriction on the channel width variation as done in [Marin and Monnier, 2009]. And our coupling term, $\Phi = (\Phi^A, \Phi^Q)^T$ clearly differs from theirs in* (4.33)*.*

Remark 4.2.2. *We will use the term* ***1D Saint Venant Model with Coupling Term*** *to refer to the model,* (4.31) *and* (4.32)*.*

4.2.2 Lateral Discharge Model During Flooding

To compute the lateral discharges in the channel, we use the following y-discharge equation in the 2D shallow water equations:

$$\partial_t q_y(\vec{\boldsymbol{X}}, t) + \partial_x f_x(\Pi) + \partial_y f_y(\Pi) = -gH(\vec{\boldsymbol{X}}, t)\partial_y z_b(\vec{\boldsymbol{X}}) \tag{4.34}$$

where

$$\Pi = (H, q_x, q_y)^T, f_x(\Pi) = \frac{q_x q_y}{H}; \quad f_y(\Pi) = \frac{q_y^2}{H} + \frac{1}{2}gH^2.$$

Remark 4.2.3. *We will use the term* ***Channel Flow Model*** *to refer to the model,* (4.31), (4.32) *and* (4.34).

4.2.3 Summary of the Channel Flow Model for the HCM

In summary, the model for describing the channel flow in HCM is the system made of equations (4.31), (4.32) and (4.34), see figure 4.5. This is the system that is solved within the channel, while the 2D shallow water equations are solved in the floodplains.

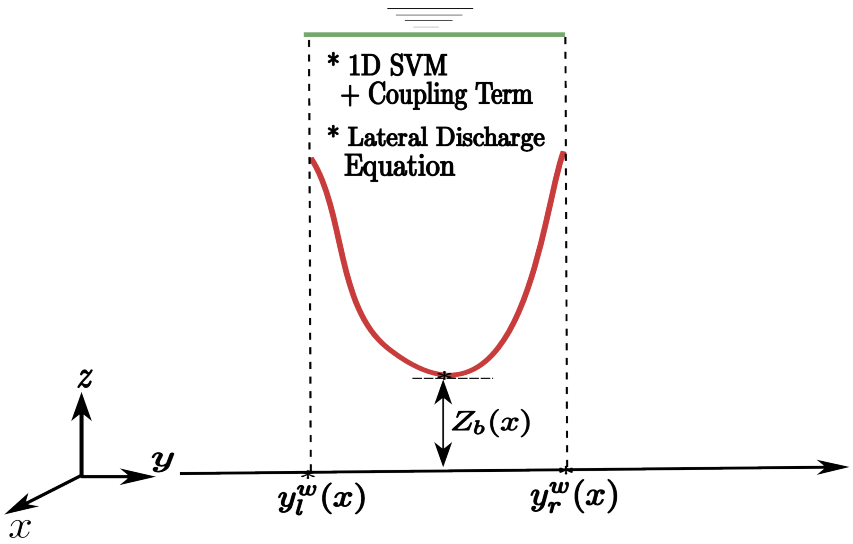


Figure 4.5: Diagram explaining the equations solved in the channel. The equations are the 1D Saint Venant model with coupling terms (4.31) and (4.32) and the lateral discharge equation (4.34). Dotted lines indicate the end of channel region at which the lateral fluxes between the channel and floodplains are computed.

4.3 Floodplain Flow Model

As mentioned above, we describe the flow in the floodplain Ω_f (see figure 4.1) using the 2D shallow water equations (2.32) presented in section 2.3.

4.4 Summary of all Models in the HCM

To conclude the modelling aspect of the HCM, we summarise that the 1D Saint Venant model with coupling term, (4.31)-(4.32) and the lateral discharge equation (4.34) are solved for the flow in the channel while the 2D shallow water flow model (2.32) is solved in the floodplains. This is depicted in figure 4.6.

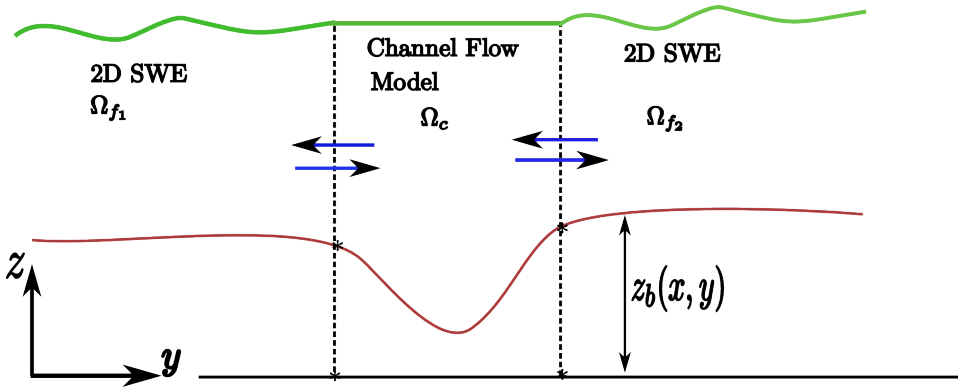


Figure 4.6: Summary of the different models for the different regions of the flow domain. The 1D Saint Venant model with coupling terms, (4.31) and (4.32) and the lateral discharge equation (4.34) are solved for the channel flow while the 2D shallow water system (2.32) is solved for the flow in the floodplains. The dotted lines indicate the boundaries between the sub domains, the point at which lateral fluxes are computed and the blue arrows indicate the presence of flow exchange between them.

This concludes the modeling aspect of the HCM. In the following sections, we focus on the numerical implementation of the HCM.

4.5 Numerical Implementation of the HCM

In this section, we detail the numerical schemes for the models presented in previous sections. Since the flood model is the 2D shallow water system whose numerical scheme has been presented in chapter 3, we focus on the numerical scheme for the channel flow model (4.31), (4.32) and (4.34). To begin, we partition the channel, Ω_c (see figure 4.1) into a 1D grid, Ω_h^{1D} made of cross sections and also partition

the floodplains, Ω_f into a 2D grid, Ω_h^{2D} and require the following: (i) The 2D grid must match with the 1D grid. This means that there should be no gap between the two grids. (ii) Each 1D channel cell should be adjacent to a whole number of 2D cells. This number can differ on either side of the 1D cell. See figure 4.7 for the illustration.

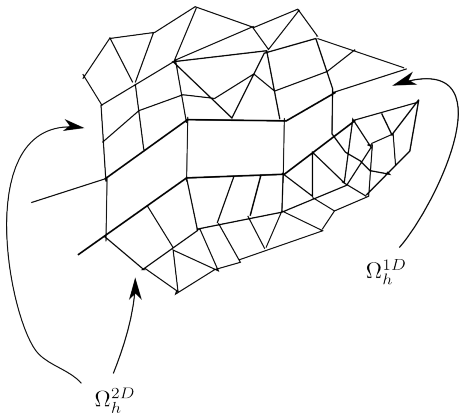


Figure 4.7: Grid of the entire domain consisting of the 1D grid Ω_h^{1D} at the middle and the 2D grids Ω_h^{2D} for the floodplains. The grids are matching in the sense that there is no gap between the 1D and the 2D grids.

4.5.1 Numerical Scheme for the Channel Flow Model

Here, we describe the finite volume method to discretize the channel flow model, equations (4.31), (4.32) and (4.34). In order to design a method that reuses existing 1D channel solvers, we discretize the 1D SVM with coupling term, (4.31) and (4.32) separately from the lateral discharge model, (4.34). And for the same reason, we will assume that we already have the discretization of the 1D SVM without the coupling term, and then discretize only the coupling term. Before delving into the details of the discretizations, let us recall the following notation from chapter 3 which we shall use throughout the chapter.

Remark 4.5.1. *$\phi^{2D} = \phi^{2D}(w_L, w_R, \vec{n})$ denotes the 2D numerical flux across the edge between two cells, L and R, with unit normal vector, $\vec{\boldsymbol{n}}$ pointing outward from cell L to cell R. Also denote by $\phi_k^{2D}(w_L, w_R, \vec{n})$, the k-th component of the numerical flux, $\phi^{2D}(w_L, w_R, \vec{n})$.*

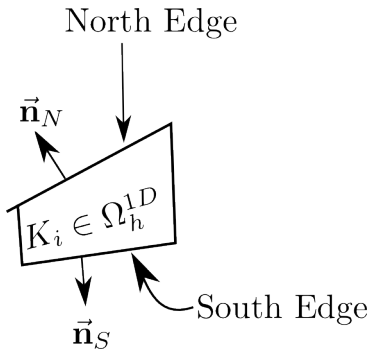


Figure 4.8: A single cell, K_i in the 1D channel mesh showing its lateral edges; South edge e_i^S is on the negative y-direction while the North edge e_i^N is on the positive y-direction. These edges are the interfaces between the 1D cell and the adjacent 2D floodplain cells.

Background

To discritize the channel flow model, let $\{x_{i+1/2}\}_{i=1}^{N_{1Dcell}}$ be points in the 1D grid, Ω_h^{1D} and $K_i = (x_{i-1/2}, x_{i+1/2}), i = 1, 2, .., N_{1Dcell}$ be a cell centred at $x_i = (x_{i-1/2} + x_{i+1/2})/2$ in Ω_h^{1D} . Where N_{1Dcell} is the number of cells in the 1D grid. Let $\boldsymbol{W}(x,t) = (A(x,t), Q(x,t))^T$ be a vector of conserved quantities at point, x and time, t, then the cell average vector, $\boldsymbol{W}_i^n = (A_i^n, Q_i^n)^T$ in cell K_i is defined as

$$\boldsymbol{W}_i^n = \frac{1}{\Delta x_i} \int_{K_i} \boldsymbol{W}(x, t^n) dx, \tag{4.35}$$

where $\Delta x_i = x_{i+1/2} - x_{i-1/2}, t^n = t^{n-1} + \Delta t$, and Δt is the time step.

For each 1D cell, $K_i \in \Omega_h^{1D}$, the channel lateral boundaries, $y = y_l^w(x)$ and $y = y_r^w(x)$ are approximated with straight edges which we call South (S) and North (N) edges (or faces) respectively with unit normals $\vec{\boldsymbol{n}}_S = (n_S^x, n_S^y)^T$ and $\vec{\boldsymbol{n}}_N = (n_N^x, n_N^y)^T$ (see figure 4.8). This means that the channel normals, $\vec{\boldsymbol{n}}_l$ and $\vec{\boldsymbol{n}}_r$ are approximated with the edge normals $\vec{\boldsymbol{n}}_S$ and $\vec{\boldsymbol{n}}_N$ respectively, that is

$$\vec{\boldsymbol{n}}_l \simeq \vec{\boldsymbol{n}}_S, \quad \vec{\boldsymbol{n}}_r \simeq \vec{\boldsymbol{n}}_N. \tag{4.36}$$

4.5.2 1D SVM with Coupling Term

In order to discretize the 1D SVM with coupling term (4.31) and (4.32), let us assume that $\boldsymbol{W}_i^{n+1*}$ is the first-order approximate solution of the 1D SVM without

the coupling term. That is, $\boldsymbol{W}_i^{n+1*}$ is the approximate solution of the system:

$$\begin{aligned} &\partial_t A + \partial_x Q = 0, \\ &\partial_t Q + \partial_x \left(\frac{Q^2}{A} \right) = -gA\partial_x \bar{\eta}(x,t). \end{aligned} \tag{4.37}$$

with the initial data $\boldsymbol{W}_i^n$. Note that $\boldsymbol{W}_i^{n+1*}$ is directly computable using the 1D scheme described in section 3.5. Also, let Φ_i^n be the discrete coupling term (derived in section 4.5.4), then the approximate solution $\boldsymbol{W}_i^{n+1}$ of the 1D SVM with the coupling term (4.31) and (4.32) is

$$\boldsymbol{W}_i^{n+1} = \boldsymbol{W}_i^{n+1*} + \Delta t \Phi_i^n. \tag{4.38}$$

To compute $\boldsymbol{W}_i^{n+1}$ in (4.38), we need $\boldsymbol{W}_i^{n+1*}$ and Φ_i^n. But since $\boldsymbol{W}_i^{n+1*}$ is to be provided by an existing 1D solver, here called "1D black box" solver, we only concern ourselves with the discretization of the coupling term, Φ_i^n. Before doing that, let us first describe the numerical scheme for the lateral discharge model, equation (4.34).

4.5.3 Approximating Channel Lateral Discharge

Consider the 1D cell, $K_i \in \Omega_h^{1D}$, with cell average vector $\boldsymbol{W}_i^n = (A_i^n, Q_i^n)^T$ from which we obtain cell average depth, $\underline{H}$ and section-averaged velocity, $\underline{u}$ as

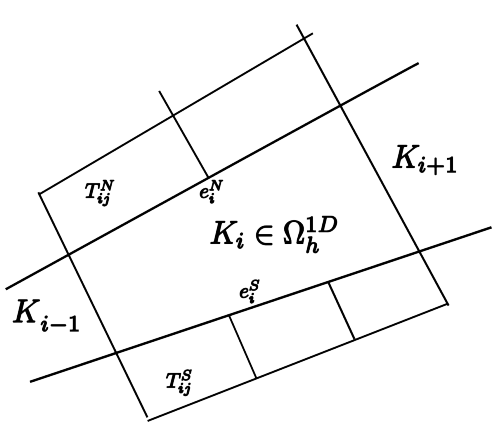
$$\underline{H}_i^n := \mathcal{H}(A_i^n, Z_{b,i}) \text{ and } \underline{u}_i^n = \frac{Q_i^n}{A_i^n} \tag{4.39}$$

respectively. Denote by $T_{ij}^N \in \Omega_h^{2D}, j = 1, 2, ..., N_n$ and $T_{ij}^S \in \Omega_h^{2D}, j = 1, 2, ..., N_s$ the j-th 2D cells adjacent to K_i on its North edge e_i^N and South edge e_i^S respectively, see figure 4.9(a). Let the cell averages in the adjacent 2D cells T_{ij}^N and T_{ij}^N be

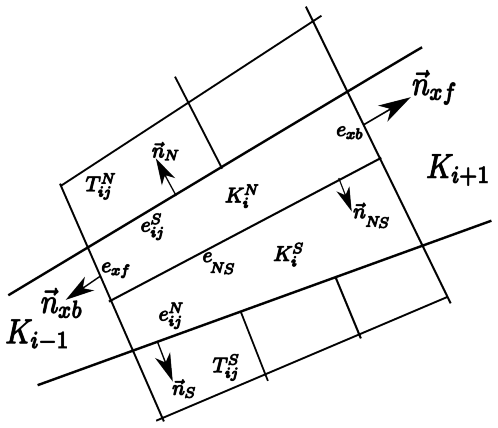
$$\begin{aligned} &(\Pi^N)_{i,j}^n = ((H^N)_{i,j}^n, (q_x^N)_{i,j}^n, (q_y^N)_{i,j}^n)^T \\ \text{and} \qquad &(\Pi^S)_{i,j}^n = ((H^S)_{i,j}^n, (q_x^S)_{i,j}^n, (q_y^S)_{i,j}^n)^T \end{aligned} \tag{4.40}$$

respectively. N_n and N_s are the number of 2D cells adjacent to K_i on the North and South edges respectively. $(H^N)_{ij}^n, (q_x^N)_{ij}^n$ and $(q_y^N)_{ij}^n$ are the average water depth, average discharge along x-direction and average discharge along y-direction respectively, in 2D cell T_{ij}^N while $(H^S)_{ij}^n, (q_x^S)_{ij}^n$ and $(q_y^S)_{ij}^n$ are those of cell T_{ij}^S.

To discretize the lateral discharge model, (4.34) we subdivide the 1D cell, K_i into two subcells, K_i^N and K_i^S and view them as 2D cells in Ω_h^{2D}, see figure 4.9(b).



(a) A 1D cell K_i in the channel grid. e_i^N and e_i^S are the North and South edges respectively. $T_{ij}^N, j = 1, 2..., N_n$ are its adjacent/neighbour 2D floodplain cells on the North edge, while $T_{ij}^S, j = 1, 2, ..., N_s$ are the adjacent 2D floodplain cells on the South edge. K_{i-1}, K_{i+1} are the left and right neighbours of K_i in 1D channel grid, Ω_h^{1D}.



(b) The single 1D cell subdivided into two subcells K_i^N and K_i^S which are then viewed as 2D cells. e_{xf}, e_{NS}, e_{xb} and $e_{ij}^N, j = 1, 2, ...N_n$ are the edges of K_i^N with their outward unit normal vectors as indicated. Similarly, the edges of K_i^S and their normal vectors are indicated.

Figure 4.9: To the left is a 1D channel cell and its adjacent 2D floodplain cells while to the right is the 1D cell subdivided into two subcells viewed as 2D cells

Let us denote by $(\boldsymbol{W}^N)_i^n$ and $(\boldsymbol{W}^S)_i^n$, the 2D cell average vectors in the subcells, K_i^N and K_i^S respectively (see figure 4.9(b)). Then, define them as

$$(\boldsymbol{W}^N)_i^n = (\underline{H}_i^n, \underline{H}_i^n \underline{u}_i^n, (q_y^N)_i^n)^T \text{ and } (\boldsymbol{W}^S)_i^n = (\underline{H}_i^n, \underline{H}_i^n \underline{u}_i^n, (q_y^S)_i^n)^T \tag{4.41}$$

where $\underline{H}_i^n, \underline{u}_i^n$ are given in (4.39) and $(q_y^S)_i^n, (q_y^N)_i^n$ are computed at every time step as explained below. Their initial values are obtained as explained in the following remark.

Remark 4.5.2. *At initial time ($n = 0$), only the lateral discharge, $(q_y)_i^0$ for the full cell, K_i is given. Then we initialize $(q_y^N)_i^0$ and $(q_y^S)_i^0$ to be equal to it, namely*

$$(q_y^N)_i^0 = (q_y^S)_i^0 = (q_y)_i^0 \tag{4.42}$$

while for other times, ($n > 0$), we compute $(q_y^{S/N})_i^n$ using the scheme described below.

Define the following:

$$h_{2ij}^N = max(0, \bar{\eta}_i^n - z_{b,ij}^N), \tag{4.43}$$

$$(\tilde{w}^N)_i^n = (h_{2ij}^N, h_{2ij}^N \underline{u}_i^n, h_{2ij}^N (v^N)_i^n)^T, \tag{4.44}$$

$$(v^N)_i^n = (q_y^N)_i^n / \underline{H}_i^n, \tag{4.45}$$

where $\bar{\eta}_i^n = \underline{H}_i^n + Z_{b,i}$ is the discrete flat free surface elevation in 1D cell, K_i and $z_{b,ij}^N$ is the bed elevation of the j-th 2D cell T_{ij}^N adjacent to subcell K_i^N, see figure 4.9(b).

The motivation to compute q_y^N is to apply a well-balanced scheme to the subcell K_i^N by taking the bottom to be flat across all the edges within the channel, e_{xb}, e_{NS}, e_{xf} (see figure 4.9(b)). Therefore, we propose the following hydrostatic reconstruction scheme [Audusse et al., 2004] for the lateral discharge in subcell K_i^N:

$$\begin{aligned}(q_y^N)_i^{n+1} = (q_y^N)_i^n - \frac{\Delta t}{|K_i^N|}\Big[& |e_{xb}| \phi_3^{2D}((\boldsymbol{W}^N)_i^n, (\boldsymbol{W}^N)_{i-1}^n, \vec{n}_{xb}) \\ & + |e_{xf}| \phi_3^{2D}((\boldsymbol{W}^N)_i^n, (\boldsymbol{W}^N)_{i+1}^n, \vec{n}_{xf}) \\ & + |e_{NS}| \phi_3^{2D}((\boldsymbol{W}^N)_i^n, (\boldsymbol{W}^S)_i^n, \vec{n}_{NS})\Big] \\ & - \frac{\Delta t}{|K_i^N|} \sum_{j=1}^{Nn} |e_{ij}^N| \Big[\phi_3^{2D}((\tilde{\boldsymbol{W}}^N)_i^n, (\Pi^N)_{ij}^n, \vec{n}_N) \\ & + \frac{g}{2} \vec{n}_N \cdot \begin{pmatrix} 0 \\ (\underline{H}_i^n)^2 - (h_{2ij}^N)^2 \end{pmatrix} \Big].\end{aligned} \tag{4.46}$$

where $\phi_3^{2D}(w_L, w_R, \vec{n})$ denotes the 3rd component of numerical flux, $\phi^{2D}(w_L, w_R, \vec{n})$ (see remark 4.5.1). $|e_{xb}|, |e_{xf}|, |e_{NS}|$ and $|e_{ij}^N|$ are the lengths of the corresponding edges of K_i^N (see figure 4.9(b)).

Similarly, we propose the following scheme for the lateral discharge in K_i^S:

$$
\begin{aligned}
(q_y^S)_i^{n+1} = (q_y^S)_i^n - \frac{\Delta t}{|K_i^S|}\Big[& |e_{xb}|\phi_3^{2D}((\boldsymbol{W}^S)_i^n, (\boldsymbol{W}^S)_{i-1}^n, \vec{n}_{xb}) \\
& + |e_{xf}|\phi_3^{2D}((\boldsymbol{W}^S)_i^n, (\boldsymbol{W}^S)_{i+1}^n, \vec{n}_{xf}) \\
& + |e_{SN}|\phi_3^{2D}((\boldsymbol{W}^S)_i^n, (\boldsymbol{W}^N)_i^n, \vec{n}_{SN})\Big] \\
& - \frac{\Delta t}{|K_i^S|}\sum_{j=1}^{Ns}|e_{ij}^S|\Big[\phi_3^{2D}((\tilde{\boldsymbol{W}}^S)_i^n, (\Pi^S)_{ij}^n, \vec{n}_S) \\
& + \frac{g}{2}\vec{n}_N \cdot \begin{pmatrix} 0 \\ (\underline{H}_i^n)^2 - (h_{2ij}^S)^2 \end{pmatrix}\Big].
\end{aligned}
\tag{4.47}
$$

where

$$h_{2ij}^S = max(0, \bar{\eta}_i^n - z_{b,ij}^S), \tag{4.48}$$

$$(\tilde{w}^S)_i^n = (h_{2ij}^S, h_{2ij}^S \underline{u}_i^n, h_{2ij}^S (v^S)_i^n)^T, \tag{4.49}$$

$$(v^S)_i^n = (q_y^S)_i^n / \underline{H}_i^n, \tag{4.50}$$

where $|e_{SN}|$ is length of edge, e_{SN} between K_i^N and K_i^S, and $z_{b,ij}^S$ is the bed elevation of 2D cell, T_{ij}^S.

4.5.4 Discrete Coupling Terms

To discretize the coupling term, let us denote by $f_{i,j}^S$ the approximation of a function, f at the edge, e_{ij}^S between 1D cell K_i and 2D cell T_{ij}^S (see figures 4.9(a) and 4.9(b)). Then the approximation, $f|_L$ of f over the entire South edge, e_i^S of K_i, is given by averaging over all edges on the South edge, namely

$$(f|_L)_i = \sum_{j=1}^{Ns}\left(f_{i,j}^S \frac{|e_{ij}^S|}{|e_i^S|}\right), \tag{4.51}$$

where $|e_{ij}^S|$ is the length of edge, e_{ij}^S; $|e_i^S|$ is sum of all edges of K_i on the South side. Similarly $(f|_R)_i = \sum_{j=1}^{Nn}\left(f_{i,j}^N \frac{|e_{ij}^N|}{|e_i^N|}\right)$ for North side, where $|e_{ij}^N|$ is the length of edge between K_i and T_{ij}^N, and $|e_i^N|$ is the sum of all edges on North side of K_i. Hence we can approximate the coupling term,

$$\Phi_i^n = (\Phi_R)_i + (\Phi_L)_i \tag{4.52}$$

as follows :

$$(\Phi_L)_i = \sum_{j=1}^{Ns} \Psi_{i,j}^S, \quad (\Phi_R)_i = \sum_{j=1}^{Nn} \Psi_{i,j}^N \tag{4.53}$$

with

$$\Psi_{i,j}^S = \begin{pmatrix} \frac{1}{n_S^y} f_{i,j}^{1,S} \\ \frac{1}{n_S^y} f_{i,j}^{2,S} - \frac{n_S^x}{n_S^y} \frac{g}{2} (H_{i,j}^{S*})^2 \end{pmatrix} \frac{|e_{ij}^S|}{|e_i^S|}, \tag{4.54}$$

$$\Psi_{i,j}^N = \begin{pmatrix} -\frac{1}{n_N^y} f_{i,j}^{1,N} \\ -\frac{1}{n_N^y} f_{i,j}^{2,N} + \frac{n_N^x}{n_N^y} \frac{g}{2} (H_{i,j}^{N*})^2 \end{pmatrix} \frac{|e_{ij}^N|}{|e_i^N|} \tag{4.55}$$

where $\Psi_{i,j}^S$ is the discrete coupling term at the edge e_{ij}^S and $\Psi_{i,j}^N$, the discrete coupling term at the edge e_{ij}^N (see figures 4.9(a) and 4.9(b)). $H_{i,j}^{S*}, f_{i,j}^{1,S}$ and $f_{i,j}^{2,S}$ are respectively, the discrete water depth, first and second components of 2D numerical flux at edge e_{ij}^S. While $H_{i,j}^{N*}, f_{i,j}^{1,N}$ and $f_{i,j}^{2,N}$ are respectively, the water depth, first and second components of 2D numerical flux at edge, e_{ij}^N. We now focus on how to compute them.

Given the 1D cell average, $\boldsymbol{W}_i^n$ in K_i from which we obtain the cell average $(\boldsymbol{W}^S)_i^n$ in the subcell, K_i^S (using equation (4.41)). Then we directly approximate $f_{i,j}^{1,S}$ and $f_{i,j}^{2,S}$ by computing the 2D numerical flux, $\phi^{2D}((\tilde{\boldsymbol{W}}^S)_i^n, (\Pi^S)_{ij}^n, \vec{n}_S)$, at edge, e_{ij}^S (see figure 4.9(b)), namely

$$\begin{aligned} f_{i,j}^{1,S} &= \phi_1^{2D}((\tilde{\boldsymbol{W}}^S)_i^n, (\Pi^S)_{ij}^n, \vec{n}_S), \\ f_{i,j}^{2,S} &= \phi_2^{2D}((\tilde{\boldsymbol{W}}^S)_i^n, (\Pi^S)_{ij}^n, \vec{n}_S), \end{aligned} \tag{4.56}$$

where $\vec{n}_S$ is the outward unit normal of K_i^S towards T_{ij}^S .

Similarly, by using $(\tilde{\boldsymbol{W}}^N)_i^n$ and $(\Pi^N)_{ij}^n$ we approximate $f_{i,j}^{1,N}$ and $f_{i,j}^{2,N}$ using

$$\begin{aligned} f_{i,j}^{1,N} &= \phi_1^{2D}((\tilde{\boldsymbol{W}}^N)_i^n, (\Pi^N)_{ij}^n, \vec{n}_N), \\ f_{i,j}^{2,N} &= \phi_2^{2D}((\tilde{\boldsymbol{W}}^N)_i^n, (\Pi^N)_{ij}^n, \vec{n}_N), \end{aligned} \tag{4.57}$$

where $\vec{n}_N$ is the outward unit normal of K_i^N towards T_{ij}^N . The hydrostatically, reconstructed quantities, $(\tilde{\boldsymbol{W}}^N)_i^n$ and $(\tilde{\boldsymbol{W}}^S)_i^n$ are defined in equations (4.44) and (4.49) respectively.

To approximate $H_{i,j}^{S*}$ and $H_{i,j}^{N*}$, we propose a simple formula by adapting the

hydrostatic reconstruction approach [Audusse et al., 2004], namely

$$H_{i,j}^{S*} = \max(h_{2ij}^{S}, (H^S)_{i,j}^{n}), \tag{4.58}$$

$$H_{i,j}^{N*} = \max(h_{2ij}^{N}, (H^N)_{i,j}^{n}), \tag{4.59}$$

where $(H^S)_{i,j}^{n}$ and $(H^N)_{i,j}^{n}$ are the cell average water depths in 2D cells T_{ij}^{S} and T_{ij}^{N} respectively. We therefore summarise the discrete coupling term as

$$\begin{aligned}\Phi_i^n = \frac{1}{|e_i^S|}\sum_{j=1}^{Ns}|e_{ij}^S|\begin{pmatrix} \frac{1}{n_S^y}\phi_1^{2D}((\tilde{\boldsymbol{W}}^S)_i^n, (\Pi^S)_{ij}^n, \vec{n}_S) \\ \frac{1}{n_S^y}\phi_2^{2D}((\tilde{\boldsymbol{W}}^S)_i^n, (\Pi^S)_{ij}^n, \vec{n}_S) - \frac{g}{2}\frac{n_S^x}{n_S^y}\left[\max(h_{2ij}^S, (H^S)_{ij}^n)\right]^2 \end{pmatrix} \\ - \frac{1}{|e_i^N|}\sum_{j=1}^{N_n}|e_{ij}^N|\begin{pmatrix} \frac{1}{n_N^y}\phi_1^{2D}((\tilde{\boldsymbol{W}}^N)_i^n, (\Pi^N)_{ij}^n, \vec{n}_N) \\ \frac{1}{n_N^y}\phi_2^{2D}((\tilde{\boldsymbol{W}}^N)_i^n, (\Pi^N)_{ij}^n, \vec{n}_N) - \frac{g}{2}\frac{n_N^x}{n_N^y}\left[\max(h_{2ij}^N, (H^N)_{ij}^n)\right]^2 \end{pmatrix},\end{aligned} \tag{4.60}$$

$n_N^y, n_S^y \neq 0$.

Remark 4.5.3. *In this formulation,* (4.60) *the North edge or face must be in the positive direction while the south must be in the negative direction. This is so because, the North represents the boundary $y = y_l^w(x)$ which is in positive direction while the South represents $y = ylwall$ which is in negative direction.*

4.5.5 Summary of the Schemes for the Channel Flow Model

We summarise the numerical scheme for the channel flow model , (4.31), (4.32) and (4.34) are summarised as follows:

$$\begin{aligned} & \boldsymbol{W}_i^{n+1} = \boldsymbol{W}_i^{n+1*} + \Delta t \Phi_i^n \\ \text{and} \quad & (q_y^{N/S})_i^{n+1} \end{aligned}$$

- $\boldsymbol{W}_i^{n+1*}$ is the approximate solution of the 1D SVM without the coupling term, (4.37) which is computed using the 1D scheme discussed in 3.5.
- Φ_i^n is the discrete coupling term summarised in (4.60).
- Lateral discharges, $(q_y^{N/S})_i^{n+1}$ are computed by (4.46) and (4.47).

4.5.6 Summary of the Schemes for the Horizontal Coupling Method

The HCM is therefore summarised as follows.

1. **Channel Flow Scheme**
   The schemes for channel flow is summarised in section 4.5.5 above.

2. **Floodplain Flow Scheme**
   The scheme for the floodplain flow is the 2D solver presented in section 3.4. At 2D/1D edge, the 2D numerical flux is computed by using the 2D cell, $T_{ij} \in \Omega_h^{2D}$ averages and the averages obtained from the adjacent 1D subcell, (K_i^N or K_i^S) as described in equation (4.41).

4.6 Properties of the HCM

We discuss a few properties of the method in this section.

Definition 4.6.1. ***(Well Balance of Lake at rest)*** *Assuming that the existing numerical schemes for the uncoupled 1D and 2D models are well balanced with respect to lake at rest, then the coupled scheme is said to be well balanced with respect to lake at rest if the coupling term vanishes whenever the lake at rest condition holds.*

Theorem 4.6.1. *The coupling term derived in equation* (4.60) *leads to a fully well-balanced scheme with respect to lake at rest.*

Proof. Assuming that the condition of water at rest holds, then

$$\bar{\eta}_i^n = (\eta^N)_{ij}^n = (\eta^S)_{ij}^n \quad \forall j$$

where $(\eta^S)_{ij}^n$ and $(\eta^N)_{ij}^n$ are the free surface elevation in the adjacent 2D cells, T_{ij}^S and T_{ij}^N respectively. Hence,

$$h_{2,ij}^S := \max(0, \bar{\eta}_i^n - z_{b,ij}^S) = \max(0, (\eta^S)_{ij}^n - z_{b,ij}^S) = \max(0, (H^S)_{ij}^n) = (H^S)_{ij}^n,$$
$$h_{2,ij}^N := \max(0, \bar{\eta}_i^n - z_{b,ij}^N) = \max(0, (\eta^N)_{ij}^n - z_{b,ij}^N) = (H^N)_{ij}^n.$$

Therefore,

$$\max(h_{2ij}^S, (H^S)_{ij}^n) = h_{2ij}^S \text{ and } \max(h_{2ij}^N, (H^N)_{ij}^n) = h_{2ij}^N. \tag{4.61}$$

Since all velocities (and discharges) are zero, then

$$(\Pi^S)_{ij}^n = ((H^S)_{ij}^n, 0, 0)^T = (h_{2ij}^S, 0, 0)^T = (\tilde{\boldsymbol{W}}^S)_i^n.$$

Hence, by the consistency of the numerical flux, ϕ^{2D} with the physical flux, $F(.)$, we have

$$\phi^{2D}((\tilde{\boldsymbol{W}}^S)_i^n, (\Pi^S)_{ij}^n, \vec{\boldsymbol{n}}_S) = F((\tilde{\boldsymbol{W}}^S)_i^n) \cdot \vec{\boldsymbol{n}}_S = (0, n_S^x \frac{g}{2}\left[h_{2ij}^S\right]^2, n_S^y \frac{g}{2}\left[h_{2ij}^S\right]^2)^T.$$

That is

$$\phi_1^{2D}((\tilde{\boldsymbol{W}}^S)_i^n, (\Pi^S)_{ij}^n, \vec{\boldsymbol{n}}_S) = 0, \quad \phi_2^{2D}((\tilde{\boldsymbol{W}}^S)_i^n, (\Pi^S)_{ij}^n, \vec{\boldsymbol{n}}_S) = n_S^x \frac{g}{2}\left[h_{2ij}^S\right]^2. \tag{4.62}$$

Similarly,

$$\begin{aligned} (\Pi^N)_j^n &= ((H^N)_{ij}^n, 0, 0)^T = (\tilde{\boldsymbol{W}}^N)_i^n \\ \text{and } \phi^{2D}((\tilde{\boldsymbol{W}}^N)_i^n, (\Pi^N)_{ij}^n, \vec{\boldsymbol{n}}_N) &= (0, n_N^x \frac{g}{2}\left[h_{2ij}^N\right]^2, n_N^y \frac{g}{2}\left[h_{2ij}^N\right]^2)^T. \end{aligned}$$

So that

$$\phi_1^{2D}((\tilde{\boldsymbol{W}}^N)_i^n, (\Pi^N)_{ij}^n, \vec{\boldsymbol{n}}_N) = 0, \quad \phi_2^{2D}((\tilde{\boldsymbol{W}}^N)_i^n, (\Pi^N)_{ij}^n, \vec{\boldsymbol{n}}_N) = n_N^x \frac{g}{2}\left[h_{2ij}^N\right]^2. \tag{4.63}$$

Therefore, using equations (4.61)-(4.63), then the discrete coupling term in (4.60) becomes

$$\begin{aligned} \Phi_i^n = &\frac{1}{|e_i^S|}\sum_{j=1}^{Ns} |e_{ij}^S| \begin{pmatrix} 0 \\ \frac{1}{n_S^y} n_S^x \frac{g}{2}\left[h_{2ij}^S\right]^2 - \frac{g}{2}\frac{n_S^x}{n_S^y}\left[h_{2ij}^S\right]^2 \end{pmatrix} \\ &- \frac{1}{|e_i^N|}\sum_{j=1}^{Nn} |e_{ij}^N| \begin{pmatrix} 0 \\ \frac{1}{n_N^y} n_N^x \frac{g}{2}\left[h_{2ij}^N\right]^2 - \frac{g}{2}\frac{n_N^x}{n_N^y}\left[h_{2ij}^N\right]^2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \end{aligned} \tag{4.64}$$

as claimed.

□

We now introduce the concept of "No-Numerical Flooding".

Definition 4.6.2. ***(No Numerical Flooding Property)*** *We shall say that a 2D/1D coupling scheme preseves the* ***No Numerical Flooding property****, if all its coupling terms vanish whenever there is no flooding or draining.*

Theorem 4.6.2. *The scheme* (4.38), (4.60) *preserves the no numerical flooding property.*

Proof. If no flooding, then

$$\begin{aligned} &\bar{\eta}_i^n \leq z_{b,ij}^S \text{ and } \bar{\eta}_i^n \leq z_{b,ij}^N \quad \forall j. \\ &\implies h_{2ij}^N = h_{2ij}^S = 0, \quad \text{by definition.} \end{aligned}$$

Hence,

$$(\tilde{\boldsymbol{W}}^N)_i^n = (\tilde{\boldsymbol{W}}^S)_i^n = (0,0,0)^T.$$

Again, since floodplain is dry, we have

$$(H^N)_{ij}^n = (H^S)_{ij}^n = 0 \quad \forall j \implies (\Pi^N)_{ij}^n = (\Pi^S)_{ij}^n = (0,0,0)^T \quad \forall j.$$

These give the numerical fluxes:

$$\phi^{2D}((\tilde{\boldsymbol{W}}^N)_i^n, (\Pi^N)_{ij}^n, \vec{n}_N) = \phi^{2D}((\tilde{\boldsymbol{W}}^S)_i^n, (\Pi^S)_{ij}^n, \vec{n}_S) = (0,0,0)^T.$$

So all the flux terms in Φ_i^n are zero.
Finally,

$$\max(h_{2ij}^N, (H^N)_{ij}^n) = \max(h_{2ij}^S, (H^S)_{ij}^n) = \max(0,0) = 0.$$

Therefore, $\Phi_i^n = 0$. Which means that no water is gained from or lost to the floodplain as required. □

4.7 Implementation Strategy

We now present one implementation strategy for the proposed coupling method.

Initialization

At initial time, $(n = 0)$, we inititialize the solutions in 1D grid cells, $K_i \in \Omega_h^{1D}$ with cell average wetted area, A_i^0, section-averaged discharge, Q_i^0 and lateral discharge, $(q_y)_i^0$ (zero if no overflow), so that $\boldsymbol{W}_i^0 = (A_i^0, Q_i^0)^T$. From these, we compute $\underline{H}_i^0$ and $\underline{u}_i^0$, then initialise the subcells, K_i^S, K_i^N averages as

$$(\boldsymbol{W}^N)_i^0 = (\boldsymbol{W}^S)_i^0 = (\underline{H}_i^0, \underline{H}_i^0 \underline{u}_i^0, (q_y)_i^0), \quad \forall K_i \in \Omega_h^{1D}.$$

The 2D cells, $T_k \in \Omega^{2D}$ are also initialised as $\Pi_k^0 = (H_k^0, (q_x)_k^0, (q_y)_k^0)^T \quad \forall T_k \in \Omega_h^{2D}$.

Update

Given 1D solutions $\boldsymbol{W}_i^n$ in cell $K_i \in \Omega_h^{1D}$ together with its subcell solutions, $(\boldsymbol{W}^N)_i^n$ and $(\boldsymbol{W}^S)_i^n$, and also 2D solutions Π_k^n in cell $T_k \in \Omega_h^{2D}$, then we update the solutions following two stages, namely the black box solver stage and the post-processing stage.

4.7.1 Black Box Solver stage

This is the stage where we use existing 1D solver to compute the solution of the 1D SVM without the coupling term, (4.37) obtaining the intermediate solution $\boldsymbol{W}_i^{n+1*} = (A_i^{n+1*}, Q_i^{n+1*})^T \quad \forall K_i \in \Omega_h^{1D}$. We also use a 2D black box solver to directly obtain $\Pi_k^{n+1} = (H_k^{n+1}, (q_x)_k^{n+1}, (q_y)_k^{n+1})^T \quad \forall T_k \in \Omega_h^{2D}$.

4.7.2 Post-Processing Stage

This stage comprises of several steps. It starts by checking if there is overflow in cell, K_i and proceeds. The following are the steps:

1. Compute threshold height, $\beta_i = z_{b,i}^w - z_{b,i}$ where $z_{b,i}^w = z_b^w(x_i)$. Note this is only computed ones.

2. Then check if $H_i^n > \beta_i$ (That is, if channel is full.)

   (a) If yes, then compute the coupling term and the lateral discharges

   (b) Otherwise, compute the coupling term only and set lateral discharges to zero.

   Although the coupling term is zero if not flooding, step 2 is still necessary in order to further reduce the amount of computations.

3. Apply the scheme, (4.38) to update 1D solution, namely

$$\boldsymbol{W}_i^{n+1} = \boldsymbol{W}_i^{n+1*} + \Delta t \Phi_i^n.$$

   The updated lateral discharges are those computed in step 2 above, while the updated 2D solutions, Π_k^{n+1} are those computed during the black box solver stage.

A flow chart of this procedure is presented in figure 4.10.

4.8 Numerical Results

In this section, we present one test case to check the accuracy of the coupling method proposed in this chapter. The idea is to compare the result with those of a full 2D simulation and the flux-based method (FBM) of [Bladé et al., 2012]. The test problem is the river-flooding problem considered in section 3.8.3. Here, a full 2D simulation was run with a grid of 193×115 cells, while a simulation with the proposed method was run with a grid made up of 193×90 cells for the floodplain and 193×2 cells in the channel. The simulation using the FBM was also run with a grid of 193×90 cells in floodplain and 193×1 in the channel. Notice that the coupling methods (the proposed one and the FBM) are using less number of grid cells, in total, than the full 2D simulation, see second row in table 4.1.

	Grid	No. of time steps	Processor time (in seconds)
Full 2D	193×115	3,669	3,110.31
HCM	193×92	2,616	1,420.4
FBM	193×91	2,592	1,311.26

Table 4.1: Grid cells, simulation times and number of time steps after ten seconds.

Figure 4.11 displays the free surface elevation for each simulation method. We can see that both the proposed method and the FBM capture the behaviour of the full 2D simulation, and from the right end of the channel, one can also see that the proposed method approximates the full 2D result better than the FBM. This is more obvious in figure 4.12, where the free surface at the probe points are plotted for each simulation method. It thus very clear that the proposed method significantly outperforms the FBM for this test case. In terms of efficiency, we notice from table 4.1 that full 2D simulation took 3,669 time steps and 3,100.31 seconds to complete this simulation while the coupling methods took less number of time steps and more than 50% reduced time to complete the same simulation. With the above observations, we conclude that it is computationally gainful to use coupling methods instead of full 2D simulation, and that the proposed method is capable of reproducing the full 2D solutions with greater accuracy than the FBM.

Remark 4.8.1. *Numerical results showed that the proposed method outperforms the FBM due to the computation of the lateral discharge, q_y as we suggested in section 4.5.3. Indeed if q_y is set to zero for the subcells K_i^N and K^S, then the result of the proposed method coincides with that of the FBM, for straight channels like the one being discussed here. We therefore conclude that* ***an efficient calculation of the***

lateral discharge in the channel truly improves the results of a coupling method.

The above result agrees with the results obtained in section 3.8.3 which revealed that if flooding, then there are lateral variations in flow quantities inside the channel. This also suggests that it is necessary to aim to compute the 2D flow structure within the channel whenever overflowing. This forms part of the motivation for the method proposed in the next chapter. In chapter 6, we present more numerical results for the present method and for the method proposed in the next chapter.

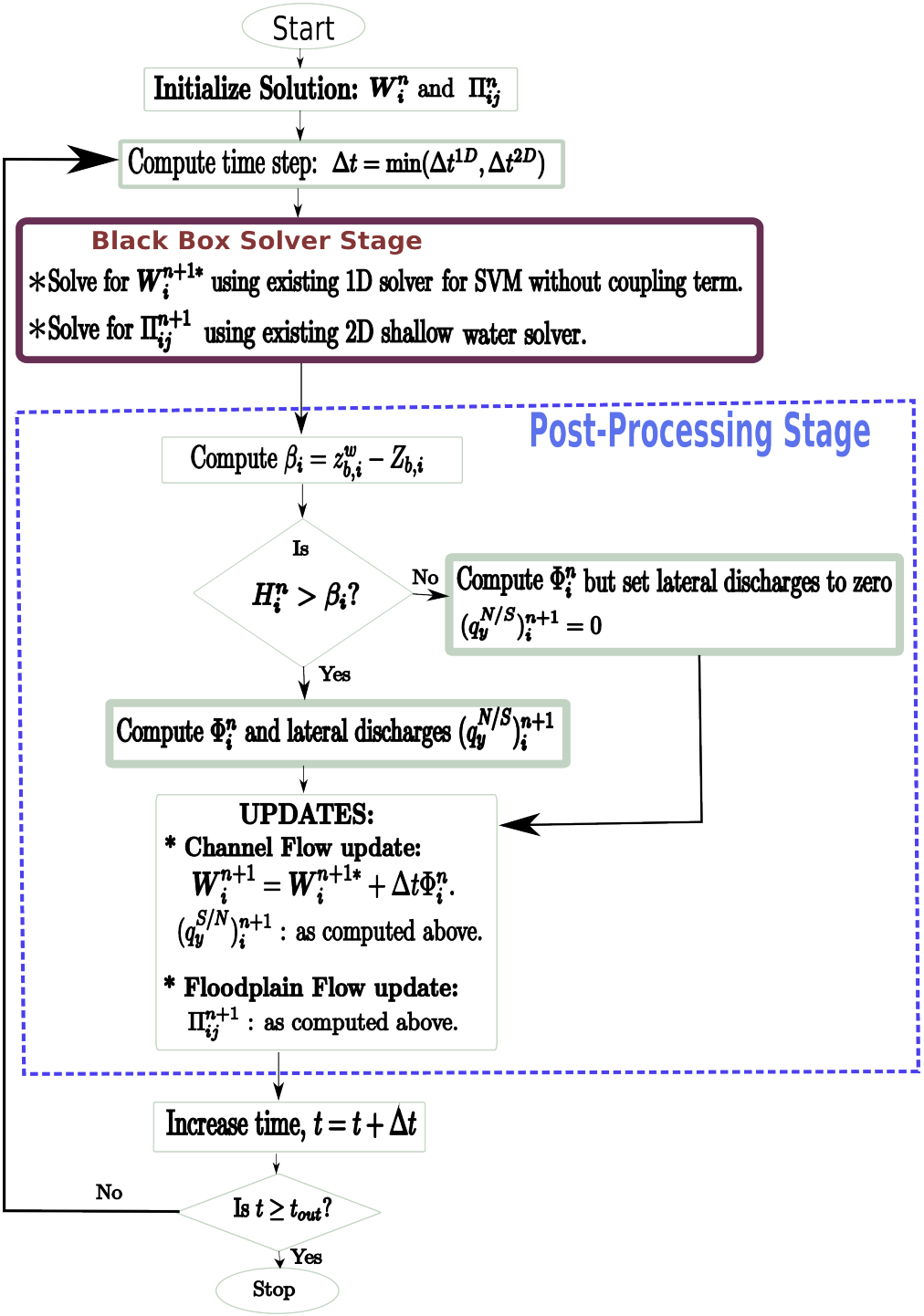


Figure 4.10: Flow Chart for horizontal Coupling Method depicting the black box solver and post processing stages. The black box solver stage computes $\boldsymbol{W}_i^{n+1*}$ using the 1D solver (3.38) and Π_{ij}^{n+1}, with the 2D solver (3.30). At the post processing stage, Φ_i^n is computed with (4.60), while $(q_y^{S/N})_i^{n+1}$ are computed using (4.46) and (4.47).

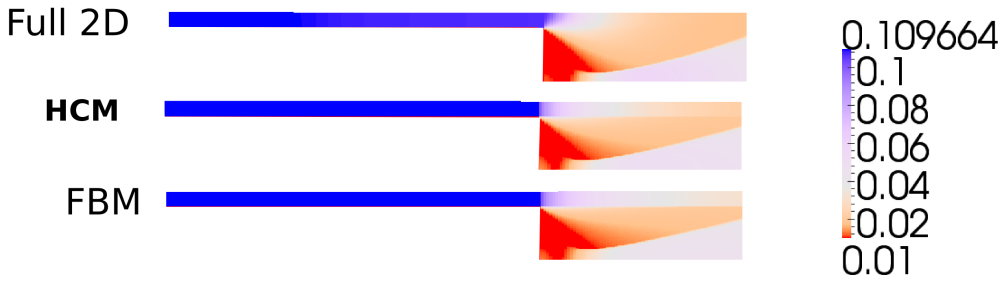


Figure 4.11: Comparison of the final free surface elevation for the HCM with those of the full 2D simulation and the FBM after ten seconds.

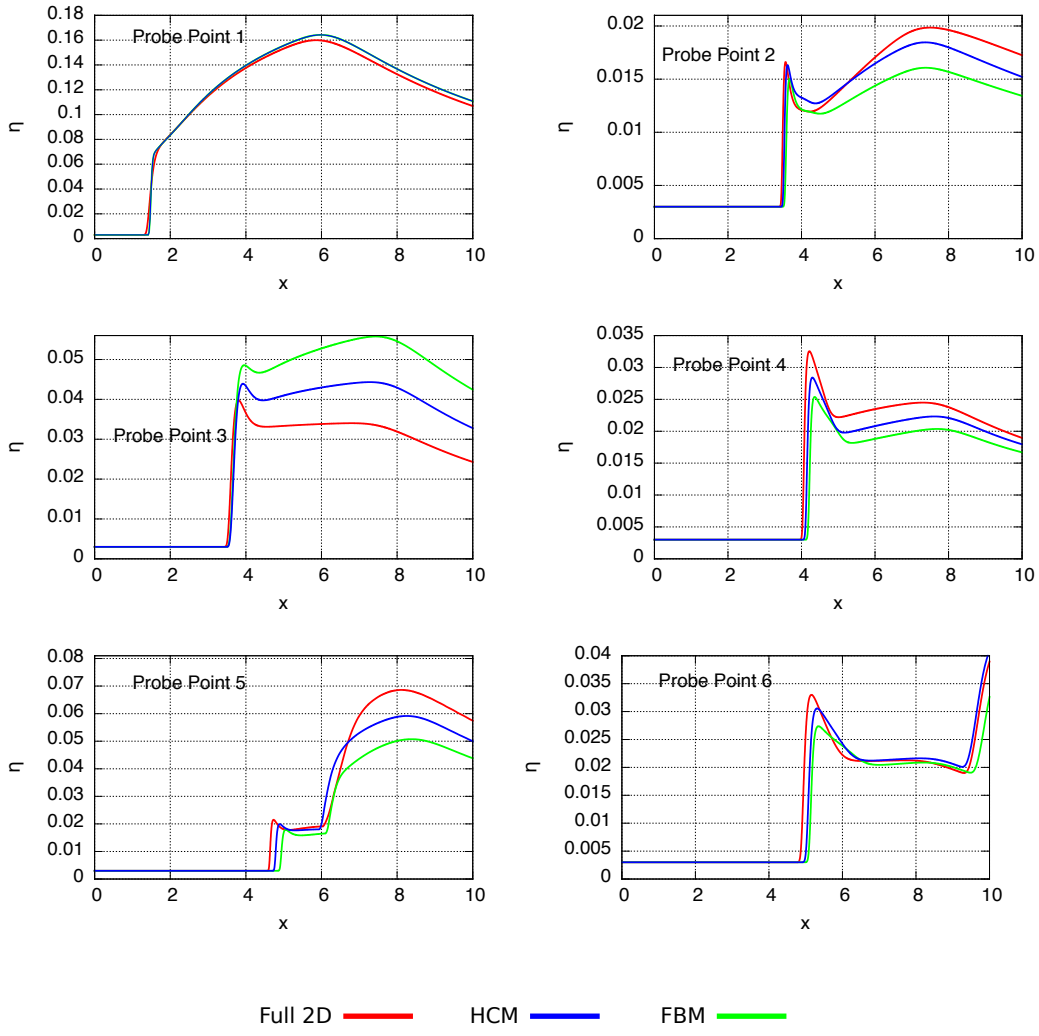


Figure 4.12: Comparison of the time evolution of the free surface elevation, η at the probe points indicated in figure 3.10.

Chapter 5

The Vertical Coupling Methods (VCM)

5.1 Introduction

The method presented in the previous chapter, and indeed most coupling methods in the literature, retain $1D$ assumptions along the channel even during flooding. In particular, they assume that the free-surface is laterally flat and discharge constant over each channel cross-section. As discussed in the previous chapter, the lateral discharge cannot be guaranteed to be constant across the channel if overflowing. In addition, the free-surface will also rarely be constant if overflowing since flow would then be activated in the lateral direction. Therefore, the assumption of constancy of lateral discharge and free-surface across channel cross-sections, is not generally realistic.

Exceptions to the methods with the above inadequate assumptions are the frontal coupling methods which compute 2D solutions at the flooding regions of the channel. However, these methods loose efficiency in the sense that they compute the 2D solutions within the channel at all times, even when flooding stops. Therefore, we need methods that self-adapt to the flooding situation; computing the true 2D solutions during flooding and reverting back to compute 1D solutions if not flooding. Again, most methods, especially the frontal coupling methods, need to know a priori the locations of flooding. But in reality, flooding locations would vary with time so might not be known a priori. Hence, methods that can automatically detect flooding regions are highly desirable.

Another point is that most methods are designed specifically for one type of flooding scenario; lateral or frontal. It is of immense importance to develop methods

that also adapt to the flooding type, being able to resolve both frontal and lateral flooding type. Consequently, it is desirable to develop coupling methods that only make $1D$ assumptions when the channel is not full, but keep the $2D$ assumptions whenever overflowing and are able to self-adapt to the different situations as listed above. This is what we set out to achieve in this chapter.

The method we propose drops the assumption of constancy of flow variables over each cross-section whenever overflowing. This method, which we call the vertical coupling method (VCM), is based on vertically partitioning the overflowing channel flow into two layers with a user defined elevation as the interface between the two layers. Although in this thesis, we only consider the case where this interface elevation is chosen to be the channel wall elevation $z_b^w(x)$, different choices of this interface lead to different schemes as discussed in section 6.1 where we explain how the VCM is a family of coupling methods. By assuming that the flow in the lower and upper layers are 1D and 2D respectively, we establish the relevant models (with exchange terms) for each layer flow. Then a three-step approach is adopted for the numerical implementation. The approach allows the possibility to recover the 2D flow structure within the channel if flooding. It also eliminates the known difficulty associated with computing the channel lateral discharges in other coupling methods.

The rest of the chapter is organised as follows. In section 5.2, we present the background including the necessary assumptions made in the method. We also explain how the entire formulation is based on an initially chosen wall elevation. We recall the fundamental equations governing the flow of fluid in such a domain in section 5.2.1. We define the cross-section layers in section 5.2.2 and derive the equations of fluid motion in the lower and upper layers in sections 5.3 and 5.4 respectively. A summary of the channel model in the VCM is given in section 5.5. A detailed presentation of the method of numerical solution is outlined in section 5.6. Let us reiterate that the main objective here is to evolve a given $2D$ flow data (in a mesh of the full 2D domain, Ω^2, see figure 4.1) from one time step to another. For the floodplains flow, we adopt the 2D SWE equations, while for the channel flow, we achieve this in section 5.6 using the derived sub-layer models and following three major steps. The first step involves distributing the given $2D$ data among the two layers. This is outlined in section 5.6.3. The second step involves updating each sub-layer data to its new time values as outlined in section 5.6.4 and finally, the third step involves combining the updated data of the two layers to update the full $2D$ data as presented in section 5.6.5.

Properties of the method are considered in section 5.7 where we prove that

the method is well-balanced, preserves no-numerical flooding and mass conservative. In tackling the problem of evolving each layer in the second step mentioned above (section 5.6.4), we adopt the operator splitting approach which involves two stages. The first stage solves the layer models without their exchange terms and also computes the lateral flow between the channel and the floodplains, while the second stage uses the solution from the first stage to evolve each layer data to their update values. It is our view that the systems of equations in the first stage of this operator splitting process, can be solved using different methods, so we choose not to fix their methods of solution but assume that their solutions (which we refer to as ***the intermediate solutions***) are given. This allowed us to concentrate on the second stage of the operator splitting process, obtaining the update values for each sub-layer in section 5.6.4. We therefore conclude section 5.6 by proposing one set of methods for solving the intermediate solutions.

5.2 Background

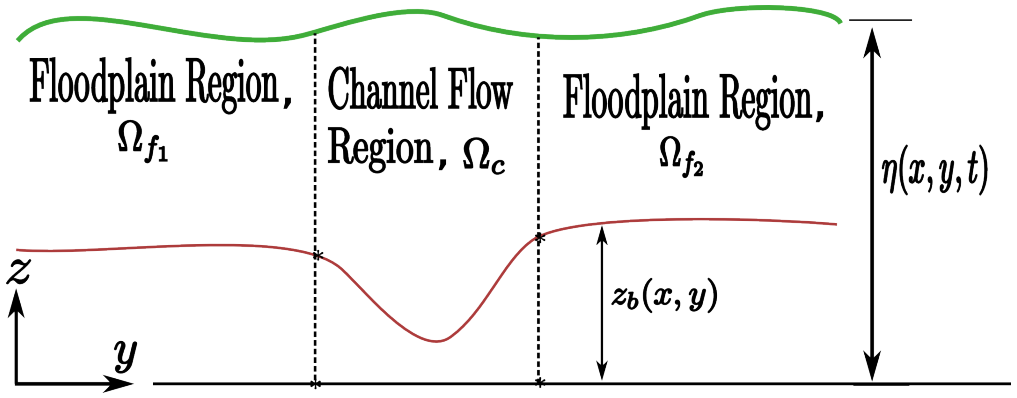


Figure 5.1: Cross sectional view of 2D flow domain, $\Omega^2 = \Omega_c \cup \Omega_f$ consisting of the river channel, Ω_c and the floodplains, $\Omega_f = \Omega_{f_1} \cup \Omega_{f_2}$. The green curve is the free surface elevation and the red curve is the bottom topography.

Just like in chapter 4, the task at hand is to couple the flow of water in the sub domains, Ω_c and Ω_f, depicted in figure 4.1 whose cross sectional view is shown in figure 4.2, see also figure 5.1. Again, we remark that the flow in the floodplain is simulated using the 2D shallow water equations like in the HCM but the channel flow model, (4.31), (4.32) and (4.34) is to be replaced by the two-layer channel flow model to be derived in the following sections. Therefore, we focus mainly on modelling the flow in the channel, Ω_c, see figure 5.2.

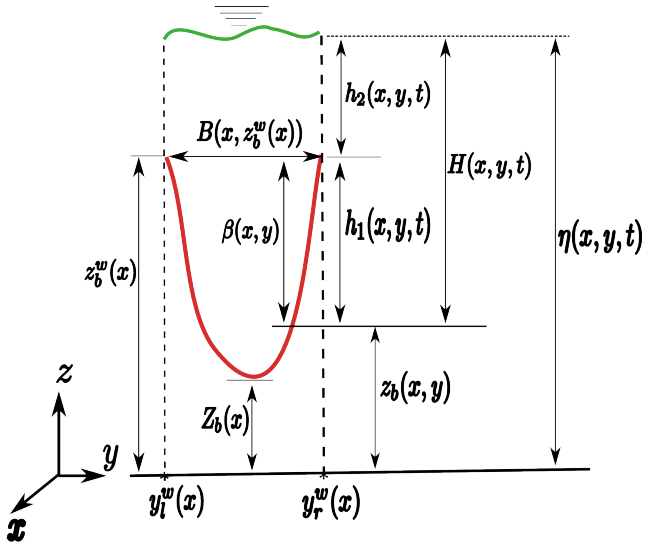


Figure 5.2: Cross section of the channel flow domain without the floodplains in the VCM. It shows the channel bottom topography $z_b(x,y)$ (in red), the channel wall elevation $z_b^w(x)$, laterally varying free-surface elevation $\eta(x,y,t)$, the bottom elevation in 1D sense $Z_b(x)$, the top width $B(x,z_b^w(x))$ and the $y-$coordinates $y_l^w(x)$ and $y_r^w(x)$ of the left and right lateral walls at the channel top. The total water depth $H(x,y,t)$, the channel depth $\beta(x,y)$, the lower layer water depth $h_1(x,y,t)$ are all measured from the same bottom elevation, $z_b(x,y)$. The upper layer water depth is $h_2(x,y,t)$.

Consider a cross section of an open channel at a fixed point, x downstream, see figure 5.2. All channel symbols : $z_b(\vec{\boldsymbol{X}}), Z_b(x), B(x,z), y_l(x,z)$, and $y_r(x,z)$ retain their meanings in figures 2.2 and 4.3.

Now, with the channel geometry and initial flow condition given, the formulation starts by choosing a wall elevation, $z_b^w(x) \geq Z_b(x)$. This elevation does not need to be equal to the maximum channel wall elevation, it can be less or greater than it, but should not be less than the 1D bottom elevation, $Z_b(x)$. Once $z_b^w(x)$ has been decided, other quantities are derived, see figure 5.2.

Then along the lateral (y) direction, the channel flow is considered in the interval; $y_l^w(x) := y_l(x,z_b^w(x)) < y < y_r^w(x) := y_r(x,z_b^w(x))$ while the floodplain flow is considered in the remaining region along that direction. This means that reducing the elevation, would reduce the lateral extent for which channel flow is considered and increase the region for 2D floodplain modelling, see figure 5.2. In particular, setting this elevation equal to 1D bottom, $Z_b(x)$ would mean to consider floodplain model everywhere, that is a full 2D simulation. So, different choices of this elevation leads to different approaches. Let us take $z_b^w(x)$ to be as previously

defined in section 2.4.1 (definition 2.4.1), that is equal to the maximum channel wall elevation as indicated in figure 5.2.

The quantity $\beta(\vec{\boldsymbol{X}})$, called the channel depth, is the laterally varying height between the channel bed and the chosen elevation, $z_b^w(x)$, that is

$$\beta(\vec{\boldsymbol{X}}) = z_b^w(x) - z_b(\vec{\boldsymbol{X}}), \qquad y_l^w(x) \leq y \leq y_r^w(x), \tag{5.1}$$

where

$$y_{l,r}^w(x) := y_{l,r}(x, z_b^w(x)). \tag{5.2}$$

We also introduce the critical area defined as follows.

Definition 5.2.1 (Critical Area)**.** *The critical area, $A_c(x)$ is the wetted cross sectional area of an exactly filled cross section. That is, the wetted cross sectional area when the free surface is flat and equal to the chosen elevation, $z_b^w(x)$. It is defined as*

$$A_c(x) = \int_{y_l(x, z_b^w(x))}^{y_r(x, z_b^w(x))} \beta(\vec{\boldsymbol{X}}) dy. \tag{5.3}$$

As can be seen in figure 5.2, the current approach, unlike the one in the previous chapter (and in the literature), allows the possibility of laterally varying flow quantities, like the free surface elevation, $\eta(\vec{\boldsymbol{X}}, t)$. Therefore, the free-surface, $\eta(\vec{\boldsymbol{X}}, t)$ now depends on y, that is, flow is $2D$ whenever the channel overflows its banks.

As proposed in section 4.2, we extend the definition of the channel width functions; $B(x, z), y_{l,r}(x, z)$ to include the overflowing region (where $z > z_b^w(x)$), in the following way,

$$\left.\begin{aligned} B(x, z) &= B(x, z_b^w(x)), \\ y_{l,r}(x, z) &= y_{l,r}(x, z_b^w(x)), \end{aligned}\right\} \quad \forall \quad z \geq z_b^w(x). \tag{5.4}$$

That is, if z is above the channel wall elevation $z_b^w(x)$, then the width is same as the channel top width.

Definition 5.2.2 (Channel Flow Lateral Boundaries)**.** *We use $y_l^*(x, t)$ and $y_r^*(x, t)$ to denote y-coordinates of the left and right channel flow lateral boundaries, given*

by

$$\begin{aligned} y_l^*(x,t) &:= \min\{y : \eta(\vec{\boldsymbol{X}},t) > z_b(\vec{\boldsymbol{X}}), y \geq y_l(x, z_b^w(x))\}, \\ y_r^*(x,t) &:= \max\{y : \eta(\vec{\boldsymbol{X}},t) > z_b(\vec{\boldsymbol{X}}), y \leq y_r(x, z_b^w(x))\}. \end{aligned} \tag{5.5}$$

We assume that the channel is never completely dry, namely $\exists y \in (y_l^w(x), y_r^w(x))$ *such that* $\eta(\vec{\boldsymbol{X}},t) > z_b(\vec{\boldsymbol{X}})$ *for all* t*. Where* $y_{l,r}^w(x) := y_{l,r}(x, z_b^w(x))$*. Note that if there is water everywhere in the channel, then* $y_{l,r}^*(x,t) = y_{l,r}^w(x) := y_{l,r}(x, z_b^w(x))$*.*

Remark 5.2.1 (Channel Assumption)**.** *We assume that the channel never goes dry anywhere between the lateral wall boundaries, namely*

$$\eta(\vec{\boldsymbol{X}},t) > z_b(\vec{\boldsymbol{X}}) \quad \forall y \in (y_l^*(x,t), y_r^*(x,t))\,.$$

Definition 5.2.3 (Average Free Surface)**.** *With the flow lateral boundaries known, we can define the laterally averaged free surface elevation, namely*

$$\bar{\eta}(x,t) = \frac{1}{B^*(x,t)} \int_{y_l^*(x,t)}^{y_r^*(x,t)} \eta(\vec{\boldsymbol{X}},t)dy, \tag{5.6}$$

where $B^*(x,t) = y_r^*(x,t) - y_l^*(x,t)$*.*

Next, we now say exactly what we mean by a full channel.

Definition 5.2.4 (Full Channel)**.** *We say the cross section is full if*

$$H(\vec{\boldsymbol{X}},t) \geq \beta(\vec{\boldsymbol{X}}) \quad \forall y \in (y_l^w(x), y_r^w(x)))\,. \tag{5.7}$$

This means that if the channel is full, then the following hold

- $$A(x,t) := \int_{y_l^w(x)}^{y_r^w(x)} H(\vec{\boldsymbol{X}},t)dy \geq \int_{y_l^w(x)}^{y_r^w(x)} \beta(\vec{\boldsymbol{X}})dy = A_c(x), \tag{5.8}$$

  where $A(x,t)$ *is the total water in the cross section.*

- $$\bar{\eta}(x,t) \geq z_b^w(x), \tag{5.9}$$

since

$$\begin{aligned}\bar{\eta}(x,t) := \frac{1}{B^*(x,t)} \int\limits_{y_l^*(x,t)}^{y_r^*(x,t)} \eta(\vec{\boldsymbol{X}},t)dy &= \frac{1}{B^*(x,t)} \int\limits_{y_l^*(x,t)}^{y_r^*(x,t)} \Big(H(\vec{\boldsymbol{X}},t) + z_b(\vec{\boldsymbol{X}})\Big)\, dy \\ &\geq \frac{1}{B^*(x,t)} \int\limits_{y_l^*(x,t)}^{y_r^*(x,t)} \Big(\beta(\vec{\boldsymbol{X}}) + z_b(\vec{\boldsymbol{X}})\Big)\, dy \\ &= \frac{1}{B^*(x,t)} \int\limits_{y_l^*(x,t)}^{y_r^*(x,t)} z_b^w(x)dy = z_b^w(x).\end{aligned}$$

Since we want to retain 1D flow assumptions if the channel is not full, then we make the following assumption to allow consistency with 1D formulations.

Definition 5.2.5 (1D Consistency Assumption)**.** *If the channel is not full, then the lateral variation in free surface elevation, $\eta(\vec{\boldsymbol{X}},t)$ is negligible and the free surface can be taken to be its lateral average, $\bar{\eta}(x,t)$. This simply means that we assume the free surface to be laterally flat. This is exactly the 1D assumption and allows to apply 1D modelling if the channel isn't full.*

Definition 5.2.6 (Time Dependent Interface)**.** *The time dependent interface, $\eta_1(x,t)$, is the elevation defined as,*

$$\eta_1(x,t) = \min(\bar{\eta}(x,t), z_b^w(x)). \tag{5.10}$$

This is the time varying interface position between the two layers that would be identified shortly.

Notice that $\eta_1(x,t) = z_b^w(x)$ if channel is full, otherwise $\eta_1(x,t) = \bar{\eta}(x,t)$. This makes the method to be self-adaptive; switching between two layer modelling (2D simulation) and one layer modelling (1D or 2D simulation). Notice again that the maximum value of $\eta_1(x,t)$ over time is $z_b^w(x)$. Hence, a change in the choice of $z_b^w(x)$ automatically changes the method. For instance, if we take $z_b^w(x)$ equal to $Z_b(x)$ (see figure 5.2), then $\eta_1(x,t) = Z_b(x)\forall t$; solving full 2D model all the time. This is the unifying nature of this method. More of this is discussed in section 6.1. Next, we show that $y_{l,r}^*(x,t) = y_{l,r}(x,\eta_1)$.

Proposition 5.2.1. $y_{l,r}^*(x,t) = y_{l,r}(x,\eta_1)$.

Proof. Case 1 : If the channel is full, then
(i) by definition 5.2.4,

$$\bar{\eta}(x,t) \geq z_b^w(x). \text{ Hence } \quad \eta_1 := min(\bar{\eta}(x,t), z_b^w(x)) = z_b^w(x).$$

(ii) By definition 5.2.2, the lateral walls are

$$y_{l,r}^*(x,t) := y_{l,r}(x, z_b^w(x)) = y_{l,r}(x, \eta_1).$$

Case 2 : If the channel is not full, then

$$\bar{\eta}(x,t) \leq z_b^w(x) \text{ so } \eta_1 := \min(\bar{\eta}(x,t), z_b^w(x)) = \bar{\eta}(x,t).$$

And definition 5.2.5 requires free surface to be constant and equal to the average, $\bar{\eta}(x,t)$, hence the free surface at the lateral walls is $\bar{\eta}(x,t)$. So, the lateral boundaries are $y_{l,r}^*(x,t) = y_{l,r}(x,\bar{\eta}) = y_{l,r}(x,\eta_1(x,t))$. □

Hence, the flow lateral walls are replaced with $y_{l,r}(x,\eta_1)$. We can now identify the channel flow domain, Ω_{ct} at time, t as:

$$\Omega_{ct} = \{(\vec{\boldsymbol{X}}, z) \in \mathbb{R}^3 : y_l(x,\eta_1(x,t)) \leq y \leq y_r(x,\eta_1(x,t)), z_b(\vec{\boldsymbol{X}}) \leq z \leq \eta(\vec{\boldsymbol{X}},t)\}. \tag{5.11}$$

In the next section, we recall the fundamental equations of fluid motion in the domain, Ω_{ct}.

5.2.1 Fundamental Equations

As presented in section 2.3, under the assumption of hydrostatic pressure, the flow in the domain, Ω_{ct} is governed by the following Free-Surface Euler Equations:

$$\nabla_{3D} \cdot \vec{\boldsymbol{U}} = 0. \tag{5.12}$$

$$\partial_t u + \nabla_{3D} \cdot (u\vec{\boldsymbol{U}}) = -g\partial_x \eta(\vec{\boldsymbol{X}}, t). \tag{5.13}$$

$$\partial_t v + \nabla_{3D} \cdot (v\vec{\boldsymbol{U}}) = -g\partial_y \eta(\vec{\boldsymbol{X}}, t). \tag{5.14}$$

in Ω_{ct} with the following kinematic boundary conditions:

$$\Big(\vec{\boldsymbol{u}} \cdot (\nabla z_b(\vec{\boldsymbol{X}})) - w\Big)\Big|_{z=z_b(\vec{\boldsymbol{X}})} = 0. \tag{5.15}$$

$$\Big(\partial_t \eta(\vec{\boldsymbol{X}},t) + \vec{\boldsymbol{u}} \cdot (\nabla \eta(\vec{\boldsymbol{X}},t)) - w\Big)\Big|_{z=\eta(\vec{\boldsymbol{X}},t)} = 0. \tag{5.16}$$

where quantities have their usual meaning as in chapter 2.2. In the next section, we partition the flow domain, Ω_{ct} into two sub-domains - lower layer and upper.

5.2.2 Defining the layers

The channel flow domain, Ω_{ct} can be decomposed into two layers by defining the following quantities:

$$H(\vec{\boldsymbol{X}}, t) = \eta(\vec{\boldsymbol{X}}, t) - z_b(\vec{\boldsymbol{X}}). \tag{5.17}$$

$$h_1(\vec{\boldsymbol{X}}, t) = \min(H(\vec{\boldsymbol{X}}, t), z_b^w(x) - z_b(\vec{\boldsymbol{X}})). \tag{5.18}$$

$$h_2(\vec{\boldsymbol{X}}, t) = H(\vec{\boldsymbol{X}}, t) - h_1(\vec{\boldsymbol{X}}, t). \tag{5.19}$$

where $H(\vec{\boldsymbol{X}}, t)$ is total water depth in the full channel cross section, $h_1(\vec{\boldsymbol{X}}, t)$ and $h_2(\vec{\boldsymbol{X}}, t)$ are the water depth in the lower and upper layers respectively, see 5.2. Hence, the sum, $h_1(\vec{\boldsymbol{X}}, t) + z_b(\vec{\boldsymbol{X}})$ defines the top of the lower layer. With our 1D consistency assumptions above, this sum, $h_1(\vec{\boldsymbol{X}}, t) + z_b(\vec{\boldsymbol{X}})$ is required to be independent of y. This is so because in the case when $h_2(\vec{\boldsymbol{X}}, t)$ is all zero in the cross- section, this sum is the free-surface and is less than or equal to $z_b^w(x)$, so must be laterally flat (independent of y). We show below that the sum, $h_1(\vec{\boldsymbol{X}}, t) + z_b(\vec{\boldsymbol{X}})$ is truly independent of y.

Lemma 5.2.1. *The sum, $h_1(\vec{\boldsymbol{X}}, t) + z_b(\vec{\boldsymbol{X}})$ is independent of y.*

Proof.

$$\begin{aligned}
\eta_1(x,t) - z_b(\vec{\boldsymbol{X}}) &:= \min(\bar{\eta}(x,t), z_b^w(x)) - z_b(\vec{\boldsymbol{X}}) \\
&= \begin{cases} z_b^w(x) - z_b(\vec{\boldsymbol{X}}), & \text{if } H(\vec{\boldsymbol{X}}, t) \geq \beta(\vec{\boldsymbol{X}}) \text{ (by definition of full cross section)} \\ \bar{\eta}(x,t) - z_b(\vec{\boldsymbol{X}}) = \eta(\vec{\boldsymbol{X}}, t) - z_b(\vec{\boldsymbol{X}}), & \text{if } H(\vec{\boldsymbol{X}}, t) < \beta(\vec{\boldsymbol{X}}) \text{ (by 1D assumption)} \end{cases} \\
&= \begin{cases} \beta(\vec{\boldsymbol{X}}), & \text{if } H(\vec{\boldsymbol{X}}, t) \geq \beta(\vec{\boldsymbol{X}}) \\ H(\vec{\boldsymbol{X}}, t), & \text{otherwise} \end{cases} \\
&= \min(H(\vec{\boldsymbol{X}}, t), \beta(\vec{\boldsymbol{X}})) = h_1(\vec{\boldsymbol{X}}, t) \qquad \text{by definition.}
\end{aligned}$$

That is,

$$h_1(\vec{\boldsymbol{X}}, t) + z_b(\vec{\boldsymbol{X}}) = \eta_1(x, t). \tag{5.20}$$

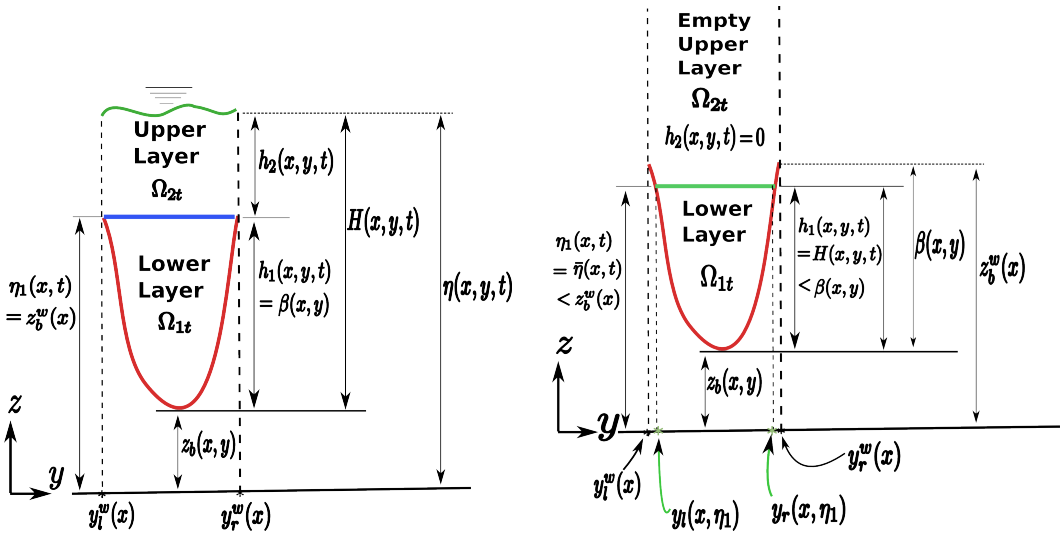
Therefore, the sum is independent of y since $\eta_1(x, t)$ is independent of y. □

We now partition the channel flow domain into the following two sub-domains

which we call layers :

$$\Omega_{1t} = \{(\vec{\boldsymbol{X}}, z) \in \mathbb{R}^3 : y_l(x, \eta_1(x,t)) \leq y \leq y_r(x, \eta_1(x,t)), z_b(\vec{\boldsymbol{X}}) \leq z \leq \eta_1(x,t)\}. \tag{5.21}$$

$$\Omega_{2t} = \{(\vec{\boldsymbol{X}}, t) \in \mathbb{R}^3 : y_l(x, \eta_1(x,t)) \leq y \leq y_r(x, \eta_1(x,t)), \eta_1(x,t) \leq z \leq \eta(\vec{\boldsymbol{X}}, t)\}. \tag{5.22}$$



(a) Full channel flow cross section depicting the lower Ω_{1t} and upper Ω_{2t} layers in the case of a full cross section. The blue line separates the two layers. Time dependent interface $\eta_1(x,t)$ is equal to the channel wall elevation $z_b^w(x)$ and lower layer water depth, $h_1(x,y,t)$ equal to the the channel depth $\beta(x,y)$. Free surface $\eta(x,y,t)$ remains non flat laterally. Channel flow boundaries $y_{l,r}(x,\eta_1)$ equal to the channel top lateral walls, $y_{l,r}^w(x)$.

(b) Non full channel flow cross section depicting the single (lower, Ω_{1t}) layer and the laterally flat free surface $\bar{\eta}(x,t)$ (in green). The time dependent interface $\eta_1(x,t)$ is equal to the laterally flat free surface $\bar{\eta}(x,t)$, which is less than the channel wall elevation $z_b^w(x)$. The lower layer water depth, $h_1(x,y,t)$ equal to the total water depth, $H(x,y,t)$ which is less than the channel depth $\beta(x,y)$. The channel flow lateral interval, $|y_r(x,\eta_1) - y_l(x,\eta_1)|$ is less than the total channel lateral interval $|y_r^w(x) - y_l^w(x)|$.

Figure 5.3: The two layers in the VCM in the case of full channel (left) and the non full case (right).

In summary, the lower layer begins at the bottom of the channel and ends at $\eta_1(x,t)$, while the upper layer is from $\eta_1(x,t)$ to the free-surface level, $\eta(\vec{\boldsymbol{X}}, t)$, see figures 5.3(a) and 5.3(b). In the next sections we formulate the equations of fluid motion in each layer, based on the assumptions, we have stated so far.

5.3 The Lower Layer Models

The lower layer is assumed to satisfy all the hypothesis of the 1D Saint Venant models (see section 2.4). This allows us to assume further that the lateral variations in free-surface elevation do not have any impact on the lower layer flow. Therefore, following the same argument in section 2.4.2, but using the averaged free-surface, $\bar{\eta}(x,t)$ we arrive at a similar reduced Free-Surface Euler Equations (see section 2.4.2), namely

$$\partial_x u(\vec{\boldsymbol{X}},z,t) + \partial_y v(\vec{\boldsymbol{X}},z,t) + \partial_z w(\vec{\boldsymbol{X}},z,t) = 0. \tag{5.23}$$

$$\begin{aligned}\partial_t u(\vec{\boldsymbol{X}},z,t) + \partial_x\left(u^2(\vec{\boldsymbol{X}},z,t)\right) + \partial_y\left(u(\vec{\boldsymbol{X}},z,t)v(\vec{\boldsymbol{X}},z,t)\right)\\ + \partial_z\left(u(\vec{\boldsymbol{X}},z,t)w(\vec{\boldsymbol{X}},z,t)\right) = -g\partial_x\bar{\eta}(x,t) = -g\partial_x\eta_1(x,t) - g\partial_x\bar{h}_2(x,t).\end{aligned} \tag{5.24}$$

in Ω_{1t}.

$$\left(u(\vec{\boldsymbol{X}},z,t)\partial_x z_b(\vec{\boldsymbol{X}}) + v(\vec{\boldsymbol{X}},z,t)\partial_y z_b(\vec{\boldsymbol{X}}) - w(\vec{\boldsymbol{X}},z,t)\right)\Big|_{z=z_b(\vec{\boldsymbol{X}})} = 0. \tag{5.25}$$

$$\left(\partial_t\bar{\eta}(x,t) + u(\vec{\boldsymbol{X}},z,t)\partial_x\bar{\eta}(x,t) - w(\vec{\boldsymbol{X}},z,t)\right)\Big|_{z=\bar{\eta}(x,t)} = 0. \tag{5.26}$$

where

$$\bar{h}_2(x,t) = \frac{1}{B(x,\eta_1)}\int_{y_l(x,\eta_1)}^{y_r(x,\eta_1)} h_2(\vec{\boldsymbol{X}},t)dy$$

5.3.1 Mass Conservation Equation for Lower Layer

Continuing with the procedure in section 2.4, we define the area of wetted cross section, $A_1(x,t)$, the section averaged volumetric discharge, $Q_1(x,t)$ and section

averaged velocity, $\underline{u}_1(x,t)$, for the lower layer, as follows :

$$Q_1(x,t) = \int_{y_l(x,\eta_1(x,t))}^{y_r(x,\eta_1(x,t))} \int_{z_b(\vec{\boldsymbol{X}})}^{\eta_1(x,t)} u(\vec{\boldsymbol{X}},z,t)dzdy, \tag{5.27}$$

$$A_1(x,t) = \int_{y_l(x,\eta_1(x,t))}^{y_r(x,\eta_1(x,t))} \int_{z_b(\vec{\boldsymbol{X}})}^{\eta_1(x,t)} dzdy = \int_{y_l(x,\eta_1(x,t))}^{y_r(x,\eta_1(x,t))} h_1(\vec{\boldsymbol{X}},t)dy \tag{5.28}$$

$$\underline{u}_1(x,t) = \frac{Q_1(x,t)}{A_1(x,t)} = \frac{1}{A_1(x,t)} \int_{y_l(x,\eta_1(x,t))}^{y_r(x,\eta_1(x,t))} \int_{z_b(\vec{\boldsymbol{X}})}^{\eta_1(x,t)} u(\vec{\boldsymbol{X}},z,t)dzdy \tag{5.29}$$

Before we proceed, let us prove the following important assertions.

Lemma 5.3.1. *Let*

$$A_c^*(x,t) = \int_{y_l(x,\eta_1(x,t))}^{y_r(x,\eta_1(x,t))} \beta(\vec{\boldsymbol{X}})dy, \tag{5.30}$$

and A_c, A be defined in (5.3) *and* (5.8) *respectively, then*

$$\min(A(x,t), A_c^*(x,t)) = \min(A(x,t), A_c(x)). \tag{5.31}$$

Proof. **Case 1:** If the channel is full, then
(i) By definition 5.2.4 ,

$$A(x,t) \geq A_c(x) \quad \text{(see equation (5.8))}.$$

(ii) By definition,

$$y_{l,r}(x,\eta_1(x,t)) = y_{l,r}^w(x).$$

Hence

$$A_c^*(x,t) := \int_{y_l(x,\eta_1(x,t))}^{y_r(x,\eta_1(x,t))} \beta(\vec{\boldsymbol{X}})dy = \int_{y_l^w(x)}^{y_r^w(x)} \beta(\vec{\boldsymbol{X}})dy = A_c(x).$$

(i) and (ii) =>

$$A_c^* = A_c \leq A => A_c = A_c^* = \min(A, A_c^*) = \min(A, A_c).$$

Case 2: If the channel is not full, then

(i) By definition,

$$A \leq A_c.$$

(ii) and

$$(y_l(x, \eta_1(x,t)), y_r(x, \eta_1(x,t))) \subset (y_l^w(x), y_r^w(x)).$$

Since $\beta(\vec{\boldsymbol{X}}) \geq 0$ for all $y \in (y_l^w(x), y_r^w(x))$, hence

$$A_c^*(x,t) := \int_{y_l(x,\eta_1(x,t))}^{y_r(x,\eta_1(x,t))} \beta(\vec{\boldsymbol{X}})dy \leq \int_{y_l^w(x)}^{y_r^w(x)} \beta(\vec{\boldsymbol{X}})dy = A_c(x).$$

(iii) By definition,

$$H(\vec{\boldsymbol{X}}, t) \leq \beta(\vec{\boldsymbol{X}}, t),$$

hence,

$$A(x,t) := \int_{y_l^w(x)}^{y_r^w(x)} H(\vec{\boldsymbol{X}}, t)dy = \int_{y_l(x,\eta_1(x,t))}^{y_r(x,\eta_1(x,t))} H(\vec{\boldsymbol{X}}, t)dy \leq \int_{y_l(x,\eta_1(x,t))}^{y_r(x,\eta_1(x,t))} \beta(\vec{\boldsymbol{X}})dy = A_c^*(x).$$

(i)-(iii) =>

$$A \leq A_c^* \leq A_c => A = \min(A, A_c^*) = \min(A, A_c).$$

Therefore, $\min(A(x,t), A_c^*(x,t)) = \min(A(x,t), A_c(x))$. as claimed □

We now prove a result very useful to numerical implementation of the VCM.

Theorem 5.3.1.

$$A_1(x,t) = \min(A(x,t), A_c(x)) \tag{5.32}$$

where A_c and A are defined as in the last lemma and A_1 is the lower layer wetted cross section defined in (5.28).

Proof. Define $B^* = y_r(x, \eta_1(x,t)) - y_l(x, \eta_1(x,t))$. Using (5.20), we can write (5.28) as follows.

$$A_1(x,t) := \int_{y_l(x,\eta_1(x,t))}^{y_r(x,\eta_1(x,t))} h_1(\vec{\boldsymbol{X}}, t)dy = \int_{y_l(x,\eta_1(x,t))}^{y_r(x,\eta_1(x,t))} (\eta_1 - z_b(\vec{\boldsymbol{X}}))dy$$

$$= B^*\eta_1 - \int_{y_l(x,\eta_1(x,t))}^{y_r(x,\eta_1(x,t))} z_b(\vec{\boldsymbol{X}})dy = B^* \min\left(\bar{\eta}(x,t), z_b^w(x)\right) - \int_{y_l(x,\eta_1(x,t))}^{y_r(x,\eta_1(x,t))} z_b(\vec{\boldsymbol{X}})dy.$$

Using the definition of $\bar{\eta}(x,t)$ in (5.6) and moving B^* under the minimum sign, we have

$$A_1(x,t) = \min\left(\int_{y_l(x,\eta_1(x,t))}^{y_r(x,\eta_1(x,t))} \eta(\vec{\boldsymbol{X}},t)dy, \int_{y_l(x,\eta_1(x,t))}^{y_r(x,\eta_1(x,t))} z_b^w(x)dy\right) - \int_{y_l(x,\eta_1(x,t))}^{y_r(x,\eta_1(x,t))} z_b(\vec{\boldsymbol{X}})dy$$
$$= \min\left(\int_{y_l(x,\eta_1(x,t))}^{y_r(x,\eta_1(x,t))} \Big(\eta(\vec{\boldsymbol{X}},t) - z_b(\vec{\boldsymbol{X}})\Big)dy, \int_{y_l(x,\eta_1(x,t))}^{y_r(x,\eta_1(x,t))} \Big(z_b^w(x) - z_b(\vec{\boldsymbol{X}})\Big)dy\right).$$

Using the definition, $H = \eta - z_b$ and $\beta = z_b^w - z_b$, then we get

$$A_1(x,t) = \min\left(\int_{y_l(x,\eta_1(x,t))}^{y_r(x,\eta_1(x,t))} H(\vec{\boldsymbol{X}},t)dy, \int_{y_l(x,\eta_1(x,t))}^{y_r(x,\eta_1(x,t))} \beta(\vec{\boldsymbol{X}})dy\right)$$
$$= \min\left(\int_{y_l^w(x)}^{y_r^w(x)} H(\vec{\boldsymbol{X}},t)dy, A_c^*(x,t)\right) = \min\Big(A(x,t), A_c^*(x,t)\Big) \qquad \text{(by definition)}$$
$$= \min\Big(A(x,t), A_c(x)\Big) \qquad \text{(by lemma 5.3.1).}$$

□

We now proceed to derive the model equations for the lower layer. As done in section 2.4.2, we integrate the equation (5.23) over the lower layer cross section, namely

$$\int_{y_l(x,\eta_1(x,t))}^{y_r(x,\eta_1(x,t))} \int_{z_b(\vec{\boldsymbol{X}})}^{\eta_1(x,t)} \Big(\partial_x u(\vec{\boldsymbol{X}},z,t) + \partial_y v(\vec{\boldsymbol{X}},z,t) + \partial_z w(\vec{\boldsymbol{X}},z,t)\Big)\,dzdy = 0. \tag{5.33}$$

Evaluating the above integral using the same procedure in section 2.4.2, (except that we do not have kinematic condition on $\eta_1(x,t)$), then we have

$$\partial_t A_1(x,t) + \partial_x Q_1(x,t) = -\int_{y_l(x,\eta_1(x,t))}^{y_r(x,\eta_1(x,t))} S(\vec{\boldsymbol{X}},t)dy, \tag{5.34}$$

where

$$S(\vec{\boldsymbol{X}},t) = \Big[w(\vec{\boldsymbol{X}},z,t) - u(\vec{\boldsymbol{X}},z,t)\partial_x\eta_1(x,t) - \partial_t\eta_1(x,t)\Big]\Big|_{z=\eta_1(x,t)}. \tag{5.35}$$

And

$$\int_{y_l(x,\eta_1(x,t))}^{y_r(x,\eta_1(x,t))} S(\vec{\boldsymbol{X}},t)dy$$

is a mass exchange term between the two layers.

5.3.2 Momentum Equation for Lower Layer

Similar to previous section, to derive the momentum equation for the lower layer, we integrate equation (5.24) over the lower layer cross section, namely

$$\begin{aligned}
&\int_{y_l(x,\eta_1(x,t))}^{y_r(x,\eta_1(x,t))} \left[\int_{z_b(\vec{\boldsymbol{X}})}^{\eta_1(x,t)} \Big(\partial_t u(\vec{\boldsymbol{X}},z,t) + \partial_y \Big(u(\vec{\boldsymbol{X}},z,t)v(\vec{\boldsymbol{X}},z,t) \Big) \right. \\
&\qquad\qquad \left. + \partial_z \Big(u(\vec{\boldsymbol{X}},z,t)w(\vec{\boldsymbol{X}},z,t) \Big) \Big) dz \right] dy \\
&= -\int_{y_l(x,\eta_1(x,t))}^{y_r(x,\eta_1(x,t))} \int_{z_b(\vec{\boldsymbol{X}})}^{\eta_1(x,t)} g\partial_x \bar{\eta}(x,t) dz dy \\
&= -gA_1(x,t)\partial_x \bar{\eta}(x,t) \\
&= -gA_1(x,t)\partial_x \eta_1(x,t) - gA_1(x,t)\partial_x \bar{h}_2(x,t).
\end{aligned}$$

Following the same procedure used deriving equation (2.66) in section 2.4.2, the above equation gives

$$\begin{aligned}
\partial_t Q_1(x,t) + \partial_x \left(\frac{Q_1^2(x,t)}{A_1(x,t)} \right) &= -gA_1(x,t)\partial_x \bar{\eta}(x,t) \\
&\quad - \int_{y_l(x,\eta_1(x,t))}^{y_r(x,\eta_1(x,t))} \Big(u(\vec{\boldsymbol{X}},z,t)|_{z=\eta_1(x,t)} S(\vec{\boldsymbol{X}},t) \Big) dy \\
&= -gA_1(x,t)\partial_x \eta_1(x,t) - gA_1(x,t)\partial_x \bar{h}_2(x,t) \\
&\quad - \int_{y_l(x,\eta_1(x,t))}^{y_r(x,\eta_1(x,t))} \Big(u(\vec{\boldsymbol{X}},z,t)|_{z=\eta_1(x,t)} S(\vec{\boldsymbol{X}},t) \Big) dy.
\end{aligned} \tag{5.36}$$

where the last integral on the right is the momentum exchange term between the two layers.

5.3.3 Summary of Lower layer Models

The complete models for the lower layer, are then

$$\partial_t A_1(x,t) + \partial_x Q_1(x,t) = -\int_{y_l(x,\eta_1(x,t))}^{y_r(x,\eta_1(x,t))} S(\vec{\boldsymbol{X}},t)dy.$$

$$\partial_t Q_1(x,t) + \partial_x\left(\frac{Q_1^2(x,t)}{A_1(x,t)}\right) = -gA_1(x,t)\partial_x\eta_1(x,t) - gA_1(x,t)\partial_x\bar{h}_2(x,t) \qquad (5.37)$$
$$- \int_{y_l(x,\eta_1(x,t))}^{y_r(x,\eta_1(x,t))} \left(u(\vec{\boldsymbol{X}},z,t)|_{z=\eta_1(x,t)} S(\vec{\boldsymbol{X}},t)\right) dy.$$

5.4 The Upper Layer Models

The upper layer flow is allowed to remain fully two-dimensional, so the Free-Surface Euler Equations, (5.12) - (5.14) and (5.16) are applicable. However, the kinematic boundary condition on the bottom does not apply here because the bottom of the upper layer is $\eta_1(x,t)$ which is not a physical boundary that fluid particles cannot cross. Following section 2.3, we define the following quantities :

$$q_{2x}(\vec{\boldsymbol{X}},t) = \int_{\eta_1(x,t)}^{\eta(\vec{\boldsymbol{X}},t)} u(\vec{\boldsymbol{X}},z,t)dz, \qquad (5.38)$$

$$q_{2y}(\vec{\boldsymbol{X}},t) = \int_{\eta_1(x,t)}^{\eta(\vec{\boldsymbol{X}},t)} v(\vec{\boldsymbol{X}},z,t)dz. \qquad (5.39)$$

So that the velocities are

$$u_2(\vec{\boldsymbol{X}},t) = \frac{q_{2x}(\vec{\boldsymbol{X}},t)}{h_2(\vec{\boldsymbol{X}},t)} \quad \text{and } v_2(\vec{\boldsymbol{X}},t) = \frac{q_{2y}(\vec{\boldsymbol{X}},t)}{h_2(\vec{\boldsymbol{X}},t)}, \qquad (5.40)$$

where $\vec{q}_2(\vec{\boldsymbol{X}},t) = (q_{2x}(\vec{\boldsymbol{X}},t), q_{2y}(\vec{\boldsymbol{X}},t))^T$ is the upper layer $2D$ discharge vector and $\vec{u}_2(\vec{\boldsymbol{X}},t) = (u_2(\vec{\boldsymbol{X}},t), v_2(\vec{\boldsymbol{X}},t))^T$, the velocity vector for upper layer. In what follows, we derive the equations for the $2D$ quantities. Integrating equation (5.12) vertically over the upper layer, we have

$$\int_{\eta_1(x,t)}^{\eta(\vec{\boldsymbol{X}},t)} \partial_x u(\vec{\boldsymbol{X}},z,t) + \partial_y v(\vec{\boldsymbol{X}},z,t) + \partial_z w(\vec{\boldsymbol{X}},z,t)dz = 0. \qquad (5.41)$$

Evaluating the integrals, the above equation gives

$$\partial_t h_2(\vec{\boldsymbol{X}},t) + \partial_x q_{2x}(\vec{\boldsymbol{X}},t) + \partial_y q_{2y}(\vec{\boldsymbol{X}},t) = S(\vec{\boldsymbol{X}},t). \tag{5.42}$$

Also, integrating equation (5.13) vertically over $\eta_1(x,t) \le z \le \eta(\vec{\boldsymbol{X}},t)$, applying the kinematic boundary condition on (equation (5.16)) and simplifying, we have

$$\begin{aligned}&\partial_t q_{2x}(\vec{\boldsymbol{X}},t) + \partial_x\left(\frac{q_{2x}^2(\vec{\boldsymbol{X}},t)}{h_2(\vec{\boldsymbol{X}},t)} + g\frac{1}{2}h_2^2(\vec{\boldsymbol{X}},t)\right) + \partial_y\left(\frac{q_{2x}(\vec{\boldsymbol{X}},t)q_{2y}(\vec{\boldsymbol{X}},t)}{h_2(\vec{\boldsymbol{X}},t)}\right)\\ &= -gh_2(\vec{\boldsymbol{X}},t)\partial_x z_b(\vec{\boldsymbol{X}}) - gh_2(\vec{\boldsymbol{X}},t)\partial_x h_1(\vec{\boldsymbol{X}},t) + u(\vec{\boldsymbol{X}},z,t)|_{z=\eta_1(x,t)}S(\vec{\boldsymbol{X}},t).\end{aligned} \tag{5.43}$$

Similarly, integrating equation (5.14) vertically over $\eta_1(x,t) \le z \le \eta(\vec{\boldsymbol{X}},t)$, applying the kinematic boundary condition on (equation (5.16)) and simplifying, we have

$$\begin{aligned}&\partial_t q_{2y}(\vec{\boldsymbol{X}},t) + \partial_x\left(\frac{q_{2x}(\vec{\boldsymbol{X}},t)q_{2y}(\vec{\boldsymbol{X}},t)}{h_2(\vec{\boldsymbol{X}},t)}\right) + \partial_y\left(\frac{q_{2y}^2(\vec{\boldsymbol{X}},t)}{h_2(\vec{\boldsymbol{X}},t)} + g\frac{1}{2}h_2^2(\vec{\boldsymbol{X}},t)\right)\\ &\qquad = -gh_2(\vec{\boldsymbol{X}},t)\partial_y z_b(\vec{\boldsymbol{X}}) + v(\vec{\boldsymbol{X}},z,t)|_{z=\eta_1(x,t)}S(\vec{\boldsymbol{X}},t),\end{aligned} \tag{5.44}$$

where $S(\vec{\boldsymbol{X}},t)$ is as defined in equation (5.35).

In summary the models for the upper layer flow are

$$\begin{aligned}\partial_t h_2 + \partial_x q_{2x} + \partial_y q_{2y} &= S(\vec{\boldsymbol{X}},t).\\ \partial_t q_{2x} + \partial_x\left(\frac{q_{2x}^2}{h_2} + \frac{g}{2}h_2^2\right) + \partial_y\left(\frac{q_{2x}q_{2y}}{h_2}\right) &= -gh_2\partial_x\left(z_b + h_1\right)\\ &\quad + u(\vec{\boldsymbol{X}},t)|_{z=\eta_1(x,t)}S(\vec{\boldsymbol{X}},t).\\ \partial_t q_{2y} + \partial_x\left(\frac{q_{2x}q_{2y}}{h_2}\right) + \partial_y\left(\frac{q_{2y}^2}{h_2} + \frac{g}{2}h_2^2\right) &= -gh_2\partial_y\left(z_b + h_1\right)\\ &\quad + v(\vec{\boldsymbol{X}},z,t)|_{z=\eta_1(x,t)}S(\vec{\boldsymbol{X}},t).\end{aligned} \tag{5.45}$$

5.5 Summary of the Channel Flow Model in the VCM

We summarise the channel flow model in the vertical coupling method as the following two coupled systems:

Lower Layer Channel Flow Model:

$$\begin{aligned}
\partial_t A_1(x,t) + \partial_x Q_1(x,t) &= -\int\limits_{y_l(x,\eta_1(x,t))}^{y_r(x,\eta_1(x,t))} S(\vec{\boldsymbol{X}},t)dy. \\
\partial_t Q_1(x,t) + \partial_x\left(\frac{Q_1^2(x,t)}{A_1(x,t)}\right) &= -gA_1(x,t)\partial_x\eta_1(x,t) - gA_1(x,t)\partial_x\bar{h}_2(x,t) \\
&\quad - \int\limits_{y_l(x,\eta_1(x,t))}^{y_r(x,\eta_1(x,t))} \left(u(\vec{\boldsymbol{X}},z,t)|_{z=\eta_1(x,t)} S(\vec{\boldsymbol{X}},t)\right) dy.
\end{aligned} \tag{5.46}$$

Upper Layer Channel Flow Model:

$$\begin{aligned}
&\partial_t h_2(\vec{\boldsymbol{X}},t) + \nabla\cdot\vec{\boldsymbol{q}}_2 = S(\vec{\boldsymbol{X}},t), \\
&\partial_t\vec{\boldsymbol{q}}_2 + \nabla\cdot F^q(h_2,\vec{\boldsymbol{q}}_2) = -gh_2\nabla(z_b+h_1) + \vec{\boldsymbol{u}}_{\eta_1} S(\vec{\boldsymbol{X}},t),
\end{aligned} \tag{5.47}$$

where

$$\begin{aligned}
&F^q = (F_2^x, F_2^y),\ F_2^x = \begin{pmatrix} \frac{q_{2x}^2}{h2} + \frac{g}{2}h_2^2 \\ \frac{q_{2x}q_{2y}}{h_2} \end{pmatrix}, F_2^y = \begin{pmatrix} \frac{q_{2x}q_{2y}}{h_2} \\ \frac{q_{2y}^2}{h_2} + \frac{g}{2}h_2^2 \end{pmatrix}, \\
&\vec{\boldsymbol{u}}_{\eta_1} = \begin{pmatrix} u(\vec{\boldsymbol{X}},z,t) \\ v(\vec{\boldsymbol{X}},z,t) \end{pmatrix}\Big|_{z=\eta_1}.
\end{aligned}$$

The above coupled systems, (5.46) and (5.47), replace the channel flow model, (4.31), (4.32) and (4.34) used in the HCM presented in the previous chapter. We make the following remarks about models derived for these two layers.

Remark 5.5.1. *We note that the models* (5.47) *and* (5.46) *are not closed since the exchange terms and interface velocities are not known. By following the idea adopted in [Audusse et al., 2011], we determine these quantities using the results of the numerical solutions of the systems without these terms. This procedure is outlined in the numerical implementation section.*

Remark 5.5.2. *The fact that our flow variables, such as* $\eta_1(x,t)$ *and* $h_1(x,y,t)$ *are defined using the minimum function means that the PDEs so derived may not make sense in the classical sense. However, our goal is not to study the mathematical properties of the PDEs for which rigorous definitions are required, but to use them to derive our desired numerical schemes. Therefore, we do not concern ourselves with the rigorous definitions, hence the above derivation has to be understood as a*

formal derivation of the governing equations under the assumption that all functions are sufficiently smooth.

In the next section, we develop the numerical procedure for the VCM.

5.6 Numerical Implementation of the VCM

5.6.1 Introduction

As stated in the introductory section above, our overall goal is to evolve a given $2D$ solution from one time instant to another. In the vertical coupling methods being proposed here, we achieve this ultimate goal through the use of the models, (5.46) and (5.47) derived in previous sections. In this section, we present the approximate solution process for the sub-layer models, (5.46) and (5.47) and show how to use their solutions to ultimately evolve the $2D$ solutions. We begin, in section 5.6.2, by defining the grids and discrete versions of the quantities appearing in the sub-models.

Consequently, the ultimate question of how to evolve a 2D solution,

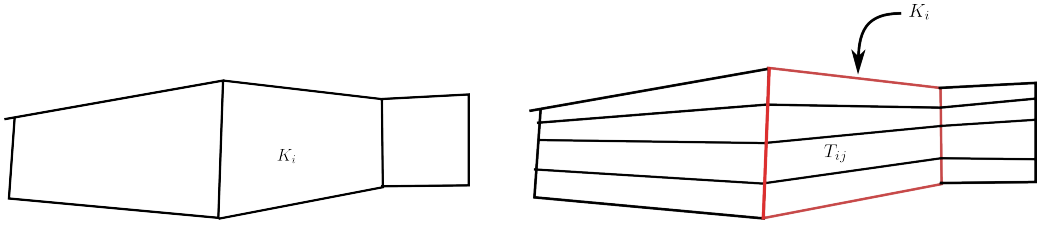
$$(H(\vec{\boldsymbol{X}}, t), q_x(\vec{\boldsymbol{X}}, t), q_y(\vec{\boldsymbol{X}}, t))^T,$$

from an old time level to a new time level within the channel is then answered by following three major steps, namely (i) Step one involves distributing the given full 2D data among the lower and upper layers. This is done in section 5.6.3. (ii) Step two involves updating each layer data to its new time values. This is detailed in section 5.6.4 and (iii) Step three involves assembling the new time values of each layer to obtain the desired full 2D data at the new time level. This is presented in section 5.6.5.

Step two above, adopts a two-stage operator splitting approach. The first stage involves solving the systems, (5.47) and (5.46) but without the exchange terms. This stage also computes the lateral flow interaction between the channel and the floodplains. The resulting approximate solutions are what we termed intermediate solutions. The second stage uses the intermediate solutions from the first stage to obtain the final update values for each layer. The solution process for the systems of equations in the first stage is not presented in section 5.6.4 but deferred to later sections. We therefore conclude the section in subsections 5.6.6-5.6.8 where we propose the methods for solving for the intermediate solutions.

5.6.2 Background

As we have severally pointed out, we solve the floodplain flow model with the 2D solver discussed in chapter 3.4. Hence, we only concern ourselves with the numerical scheme for the channel flow model, (5.46), (5.47) derived in the previous section. Therefore, we consider the discretization of the channel domain alone.



(a) Top view of 1D channel mesh made of cells K_i. The lower layer model is to be solved on each cell, K_i.

(b) Top view of 2D mesh for the channel region only - no floodplains. Each 1D channel cell K_i (in red) is further discretized to form a number, N_y^1 of 2D cells, $T_{ij}, j = 1, 2, ..., N_y^1$.

Figure 5.4: Mesh for the two sub-models.

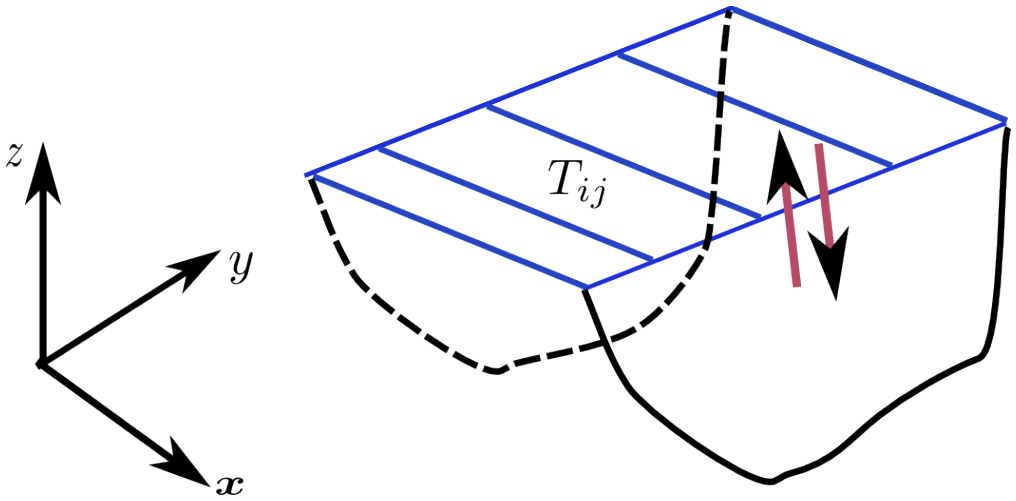


Figure 5.5: Three dimensional view of the channel meshes (without floodplain) showing a single 1D channel cell, K_i (the largest rectangle in blue) and the 2D channel cells, T_{ij} (the smaller blue rectangles). The channel bottom topography is shown in black. The arrows indicate the flow exchange between the two layers in the channel and it is computed during the second stage of the operator splitting method presented later.

Let $\{x_{i+1/2}\}_{i=0}^{N_x^1}$ be set of points along the channel and $x_i = \left(x_{i-1/2} + x_{i+1/2}\right)/2$, then we partition the channel, Ω_c (see figure 4.1) into a $1D$ mesh, Ω_h^1 made up of

cells, $K_i = (x_{i-1/2}, x_{i+1/2})$ (see figure 5.4(a)). We further discretize the channel into a 2D mesh Ω_h^2 such that each 2D cell, T_{ij} is a subcell of a unique 1D cell, $K_i \in \Omega_h^1$, that is,

$$\Omega_h^2 = \{T_{ij} : \exists! K_i \in \Omega_h^1 \text{ such that } T_{i,j} \in K_i\}, \tag{5.48}$$

see figure 5.4(b). We also discretize time as, $t^n, n = 1, 2, ...N^n$.

Solving the coupled two layer models then means to solve the 1D models (5.46) on $K_i \in \Omega_h^1$ and solve the 2D models (5.47) in the corresponding 2D cells, $T_{ij} \in K_i$. Hence the discussion in this section will be for a fixed 1D cell K_i and its sub-cells T_{ij}.

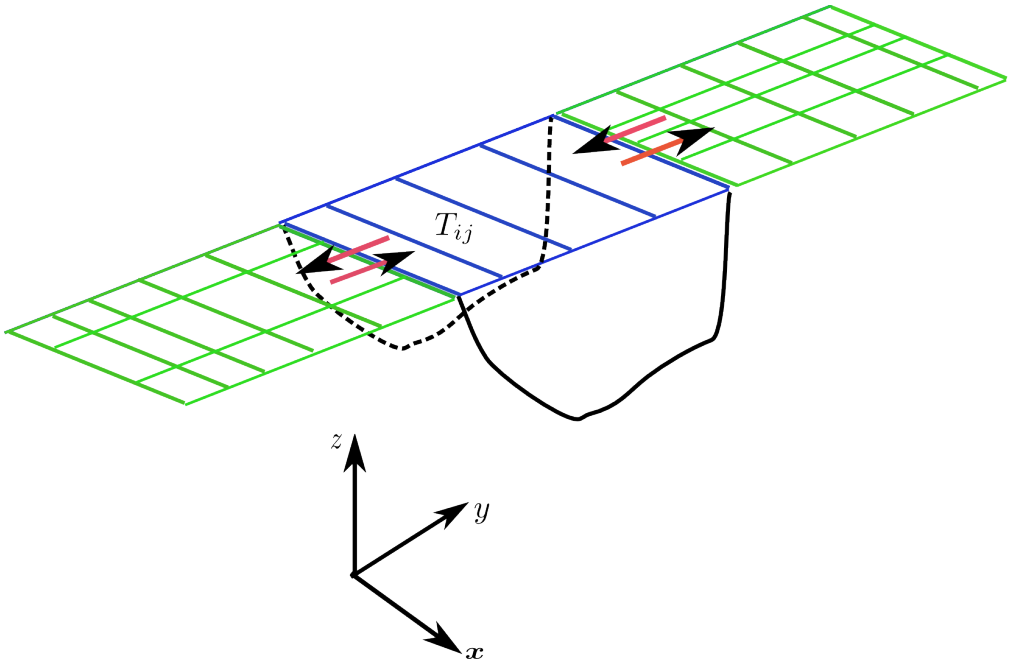
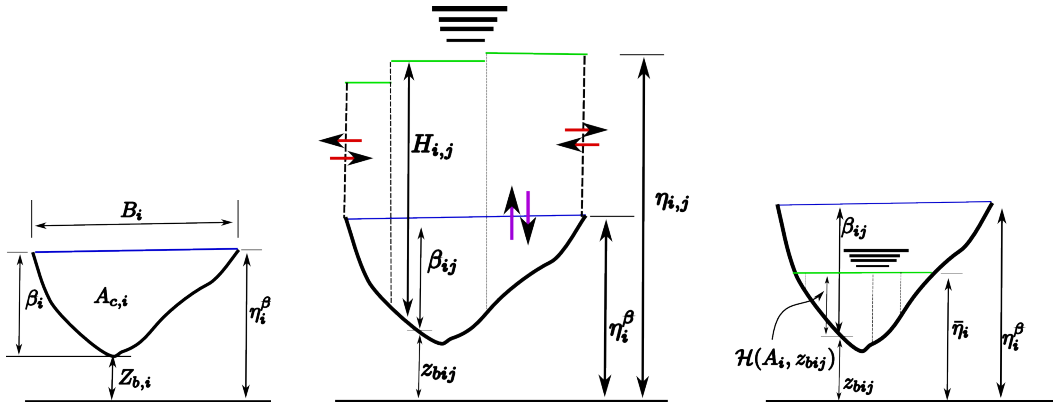


Figure 5.6: Three dimensional view of a single 1D channel cell, K_i (the largest rectangle in blue), the 2D channel cells, T_{ij} (the smaller blue rectangles) and the floodplain mesh (in green). The arrows indicate the flow exchange between the channel and the floodplain. These interactions are seamlessly computed by the 2D upper layer model while computing the intermediate solutions in the channel. The two layer model is not solved in the floodplain (green) mesh.



(a) 1D channel cross section, K_i depicting the discrete channel wall elevation, η_i^β, top width, B_i, depth β_i, critical area, $A_{c,i}$ and 1D bottom elevation $Z_{b,i}$.

(b) Full channel cross section viewed in $2D$ channel mesh. The laterally varying free-surface elevation, $\eta_{i,j}$ (in green) channel bottom elevation z_{bij}, channel depth β_{ij} and water depth $H_{i,j}$ in the 2D channel cell $T_{ij} \in \Omega_h^2$. Red arrows indicate the flow exchanges between the channel and the floodplain while the purple arrows indicate the flow exchange between the two layers of the channel flow.

(c) Non-full channel cross section viewed in $2D$ channel mesh cells. The laterally flat free-surface elevation, $\bar{\eta}_i$ which is less than the channel elevation η_i^β.

Figure 5.7: Discrete channel geometry in 1D channel cell (figure 5.7(a)) and the cross sectional view of the channel flow 2D channel cells when full (figure 5.7(b)) and non-full (figure 5.7(c)).

For the $1D$ cell, K_i, we define the following discrete channel quantities:

$$\begin{aligned}
&\eta_i^\beta = z_b^w(x_i) \quad \text{(the channel wall elevation)},\\
&B_i = B(x_i, \eta_i^\beta) \quad \text{(the channel top width)},\\
&A_{c,i} = A_c(x_i) \quad \text{(the critical area)},\\
&Z_{bi} = Z_b(x_i) \quad \text{(the 1D bed elevation)},\\
&\beta_i = \eta_i^\beta - Z_{b,i} \quad \text{(the channel depth)}
\end{aligned} \tag{5.49}$$

at the center of the 1D cell, K_i (see figure 5.7(a)). The discrete cell averages;

$$\begin{aligned} A^n_{1,i} &\approx \frac{1}{\Delta x_i} \int_{x_i-1/2}^{x_i+1/2} A_1(x,t^n)dx \\ Q^n_{1,i} &\approx \frac{1}{\Delta x_i} \int_{x_i-1/2}^{x_i+1/2} Q_1(x,t^n)dx \\ A^n_{i} &\approx \frac{1}{\Delta x_i} \int_{x_i-1/2}^{x_i+1/2} A(x,t^n)dx \end{aligned} \tag{5.50}$$

are for the wetted cross section and section-averaged discharge in lower layer and total wetted cross sectional area in cell, K_i.

For the associated $2D$ cells, T_{ij} in K_i, we define the following:

$$\begin{aligned} z_{bij} &= z_b(\vec{X}_{ij}), \\ \beta_{ij} &= \eta^\beta_i - z_{bij}, \end{aligned} \tag{5.51}$$

being the bed elevation and channel depth at the cell center, $\vec{X}_{ij}$ of T_{ij}, see figure 5.7(b). Note the identity, $Z_{bi} + \beta_i = z_{bij} + \beta_{ij} = \eta^\beta_i \quad \forall j = 1, ..., Ny^1$.
We also define the cell averages for the 2D upper layer,

$$\begin{aligned} h^n_{2,i,j} &\approx \frac{1}{|T_ij|} \int_{T_{ij}} h_2(\vec{\boldsymbol{X}}, t^n)dxdy, \\ \vec{q}^n_{2,i,j} &\approx \frac{1}{|T_ij|} \int_{T_{ij}} \vec{q}_2(\vec{\boldsymbol{X}}, t^n)dxdy, \end{aligned} \tag{5.52}$$

where $\vec{q}_2 = (q_{2x}, q_{2y})^T$ and $|T_{ij}|$ is the size(area) of $T_{ij} \in K_i$.

Finally, we also note the following,

$$H^n_{i,j} \approx \frac{1}{|T_ij|} \int_{T_{ij}} H(\vec{\boldsymbol{X}}, t^n)dxdy, \quad \vec{q}^n_{i,j} \approx \frac{1}{|T_ij|} \int_{T_{ij}} \vec{q}_(\vec{\boldsymbol{X}}, t^n)dxdy, \tag{5.53}$$

which denote the cell averages for the full 2D data (sum of lower and upper layer) in 2D cells, $T_{ij} \in \Omega^2_h$.

Notation For Water Depth in each $2D$ Cell, Given the Total Wetted Area in a Cross-Section

Given that A_i^n is the wetted cross-section area in cross-section, K_i and z_{bij} is a bottom elevation of a 2D cell in K_i, then we denote by $\mathcal{H}(A_i^n, z_{bij})$, the water depth in the $2D$ cell, $T_{ij} \in K_i$ such that the free surface is flat. That is, the function,

$$\mathcal{H} : \mathbb{R}^2 \to \mathbb{R} \tag{5.54}$$

takes the wetted cross sectional area in K_i and a bottom elevation, then returns the water depth from the given bottom elevation to the flat free surface (See figure 5.7(c)). For instance, the maximum lower layer depth, β_{ij} corresponds to the water depth when the wetted cross-section area is equal to the critical area, $A_{c,i}$, hence

$$\mathcal{H}(A_{c,i}, z_{bij}) = \beta_{ij}. \tag{5.55}$$

Remark 5.6.1. *Note that the definition of the function, $\mathcal{H}$ has to be given (from the geometry of the channel).*

Notation For Total Wetted Area in a Cross-Section, Given the $2D$ Free-Surface Elevations

Given that the lateral distribution of the free-surface elevation in the 2D cells, T_{ij} in K_i are $\eta_{i,1}^n, \eta_{i,2}^n, ..., \eta_{i,N_y^1}^n$, then we denote the wetted cross sectional area, A_i^n in K_i by

$$\mathcal{A}(\eta_{i,1}^n, \eta_{i,2}^n, ..., \eta_{i,N_y^1}^n) \quad (\equiv A_i^n). \tag{5.56}$$

So, the function,

$$\mathcal{A} : \mathbb{R}^{N_y^1} \to \mathbb{R} \tag{5.57}$$

takes all the free-surface elevations in the 2D cells in K_i and returns the wetted cross sectional area in K_i (see figure 5.7(b)). For example, we know that if all the 2D cells have flat free surface equal to the chosen wall elevation, η_i^β, then the wetted cross sectional area is the critical area, $A_{c,i}$ (see definition 5.2.1), therefore,

$$\mathcal{A}(\eta_i^\beta, \eta_i^\beta, ..., \eta_i^\beta) = A_{c,i}. \tag{5.58}$$

Furthermore, by construction, the lower layer top surface elevation, $\eta_{1,i}^n$ is always laterally flat, so is equal in all 2D cells in K_i. Hence, the following relationship between the lower layer wetted cross section area, $A_{1,i}^n$ and the lower layer top

elevation, $\eta^n_{1,i}$, is true

$$A^n_{1,i} = \mathcal{A}(\eta^n_{1,i}, \eta^n_{1,i}, ..., \eta^n_{1,i}). \tag{5.59}$$

Finally, the upper layer is rectangular, so its area, $A^n_{2,i}$ of wetted cross section is $A^n_{2,i} = \sum_{j=1}^{N^1_y} h^n_{2,i,j}\Delta y_{i,j}$. Hence, the relation $A^n_i = A^n_{1,i} + A^n_{2,i}$, leads to the following

$$A^n_i = \mathcal{A}(\eta^n_{1,i}, \eta^n_{1,i}, ..., \eta^n_{1,i}) + \sum_{j=1}^{N^1_y} h^n_{2,i,j}\Delta y_{ij} = \mathcal{A}(\eta^n_{i,1}, \eta^n_{i,2}, ..., \eta^n_{i,N^1_y}). \tag{5.60}$$

The function, $\mathcal{A}$, has to be **known** from the channel geometry. For instance, if the channel is of rectangular cross-section, with a laterally constant bottom elevation, $z_{bij} = Z_{bi}$ for all j, then $\mathcal{A}$ would be given by

$$\mathcal{A}(\eta^n_{i,1}, \eta^n_{i,2}, ..., \eta^n_{i,N^1_y}) = \sum_{j}^{N^1_y}(\eta^n_{i,j} - Z_{bi})\Delta y_{ij} = \sum_{j}^{N^1_y} H^n_{i,j}\Delta y_{ij}, \tag{5.61}$$

where Δy_{ij} is the lateral width of T_{ij}.

Remark 5.6.2. *We do not define $\mathcal{A}$ using 2D bottom elevation, z_{bij}. In other words, we do not want to calculate the lower layer area, $A^n_{1,i}$ using formulae like $\sum_{j=1}^{N^1_y} h^n_{1,i,j}\Delta y_{i,j}$, neither do we want to compute the total area A^n_i using formulae* (5.61) *except in the case of rectangular channels. Instead, we use $\mathcal{A}$ as a generalisation for areas. This is important since we want to recover 1D channel flow exactly, should the cross section not be full.*

We recall the 1D consistency assumption in discrete version.

Definition 5.6.1 (Discrete Consistency requirement)**.** *In line with the consistency requirement in definition 5.2.5, we say that the solution, $(H, q_x, q_y)^n_{i,j}$ at t^n satisfies a discrete consistency requirement, if $\exists j^* \in \{1, 2, ..., N^1_y\}$ such that $\eta^n_{i,j^*} < \eta^\beta_i$, then*

$$\eta^n_{i,j} = \bar{\eta}^n_i < \eta^\beta_i \quad \forall j \in \{1, 2, ..., N^1_y\}. \tag{5.62}$$

We will later show that if the initial data satisfies this discrete consistency requirement, then the requirement is satisfied for all time.

In the next section, we begin with the first step of the numerical implementation process, namely to distribute a given full $2D$ data, $(H, \vec{q})^n_{ij}$ among the two layers.

5.6.3 Step 1: Distribution of Full 2D data into two-layer data

Given the full 2D data, $(H, q_x, q_y)^n_{i,j}$ and z_{bij} in the 2D grid, Ω^2_h, we distribute it among the two layers as follows. Define

$$\eta^n_{i,j} = H^n_{i,j} + z_{bij}, u^n_{i,j} = \frac{q^n_{x,i,j}}{H^n_{i,j}}, \tag{5.63}$$

$$\bar{\eta}^n_i = \frac{1}{N^1_y} \sum_j^{N^1_y} \eta^n_{i,j}, \tag{5.64}$$

$$h^n_{1,i,j} = min(H^n_{i,j}, \beta_{ij}), \quad q^n_{1x,i,j} = h^n_{1,i,j} u^n_{i,j}, \tag{5.65}$$

where $\eta^n_{i,j}$ and $u^n_{i,j}$ are the cell average free surface elevation and x-velocity respectively, in 2D cell and $\bar{\eta}^n_i$ is the averaged $\eta^n_{i,j}$ for all 2D cells in the 1D cell, K_i. $h^n_{1,i,j}$ and $q^n_{1,i,j}$ are the lower layer (viewed as a 2D cell) height and x-discharge respectively. Having obtained the lower layer data ($h^n_{1,i,j}, q^n_{1,i,j}$) in 2D, we then use them to obtain the required 1D lower layer data for the 1D model.

Procedure For Lower Layer 1D Data

We initialise the lower layer data as follows.

$$\begin{aligned} \eta^n_{1,i} &:= \min\left(\bar{\eta}^n_i, \eta^\beta_i\right) = \min\left(\eta^n_{i,1}, \eta^\beta_i\right) \text{ (by definition 5.6.1).} \\ A^n_{1,i} &= \mathcal{A}(\eta^n_{1,i}, \eta^n_{1,i}, ..., \eta^n_{1,i}), \\ Q^n_{1,i} &= \sum_{j=1}^{N^1_y} q^n_{1x,i,j} \Delta y_{i,j}, \end{aligned} \tag{5.66}$$

where $\Delta y_{i,j}$ is the lateral width of $T_{i,j} \in K_i$.

Procedure For Upper Layer Data

$$\begin{aligned} h^n_{2,i,j} &= H^n_{i,j} - h^n_{1,i,j}, \\ q^n_{2x,i,j} &= h^n_{2,i,j} u^n_{i,j}, \\ q^n_{2y,i,j} &= h^n_{2,i,j} v^n_{i,j}, \end{aligned} \tag{5.67}$$

where $v^n_{i,j} = \frac{q^n_{y,i,j}}{H^n_{i,j}}$. Haven obtained the current time level data for each layer, we now use them to update each layer models in the next section, which is the second step of the algorithm.

5.6.4 Step 2: Operator Splitting Method For Sub-Layer Models

This is the section that we actually solve the two layer channel model, (5.46), (5.47) to update each layer solution. This involves two steps which we call stages. In stage one, we solve each model without the exchange terms. In this stage, the upper layer takes into account the flow exchange between the channel and the floodplains. This is easily achieved through the computation of lateral fluxes in the 2D channel cells adjacent to the floodplain cells, see figures 5.6 and 5.7(b). The second stage is where we resolve the flow exchange between the lower and upper layer flows in the channel. We adopt the operator splitting method for this two-stage process.

Step 2, stage 1: Sub-layers' intermediate solutions and channel/floodplain interaction

As mentioned above, this stage solves the channel flow model without the exchange terms and at the same time, resolves the flow exchange between the channel and the floodplains through the lateral fluxes in the upper layer channel model. To proceed, we adopt the operator splitting approach as follows: Let $(A_{1,i}, Q_{1,i})^{n+1*}$ be the approximate solution of the 1D lower layer model, (5.46) without the exchange terms, namely

$$\begin{aligned} \partial_t A_1 + \partial_x Q_1 &= 0, \\ \partial_t Q_1 + \partial_x Q_1^2/A_1 &= -gA_1\partial_x \bar{\eta}, \end{aligned} \tag{5.68}$$

with the given initial data, $(A_{1,i}, Q_{1,i})^n$. Also, let $(h_2, \vec{q}_2)_{i,j}^{n+1*}$ be the approximate solution of the upper layer 2D models,(5.47), without the exchange terms, namely

$$\begin{aligned} \partial_t h_2 + \nabla.\vec{q}_2 &= 0, \\ \partial_t \vec{q}_2 + \nabla.F^q(h_2, \vec{q}_2) &= -gh_2\nabla(z_b + h_1), \end{aligned} \tag{5.69}$$

given the initial data $(h_2, \vec{q}_2)_{i,j}^n$. $F^q(h_2, \vec{q}_2) = (F_2^x, F_2^y)$, with $F_2^x = (\frac{q_{2x}^2}{h2} + \frac{g}{2}h_2^2, \frac{q_{2x}q_{2y}}{h_2})^T$, $F_2^y = (\frac{q_{2x}q_{2y}}{h_2}, \frac{q_{2y}^2}{h_2} + \frac{g}{2}h_2^2,)^T$. Recall that solving the above upper layer system, (5.69) involves computing the fluxes across all cell faces, in particular, the lateral fluxes between the channel and the floodplains are computed in obtaining $(h_2, \vec{q}_2)_{i,j}^{n+1*}$ for all channel 2D cells, T_{ij} adjacent to the floodplains. This way, the flow interaction between the channel and the floodplains are resolved.

In the remainder of this chapter, we will refer to the solutions of the sub-models without the exchange terms, (5.68) and (5.69), namely $(A_{1,i}, Q_{1,i})^{n+1*}$ and $(h_2, \vec{q}_2)_{i,j}^{n+1*}$ as the ***intermediate solutions***. At the moment, we assume that

they (***the intermediate solutions***) have been obtained using an appropriate PDE solver. We postpone their solution methods to sections 5.6.6-5.6.8.

Step2, stage 2: Sub-layers' Exchange and Update

In this step (stage 2), we resolve the exchange between the two layers and update each layer. Suppose we have the intermediate solutions, $(A_{1i}, Q_{1i})^{n+1*}$ and $(h_2, \vec{q}_2)_{i,j}^{n+1*}$, then the approximate solution of the 1D lower layer models including the exchange terms, (5.46) is the approximate solution of the system

$$\begin{aligned} \partial_t A_{1,i} &= -\int S dy, \\ \partial_t Q_{1,i} &= -\int u_{\eta_1} S dy, \end{aligned} \tag{5.70}$$

given the initial data, $(A_{1,i}, Q_{1,i})^{n+1*}$, and the approximate solution of the 2D models, (5.47) including the exchange terms, is the approximate solution of the system

$$\begin{aligned} \partial_t h_{2,i,j} &= S, \\ \partial_t \vec{q}_{2,i,j} &= \vec{\boldsymbol{u}}_{\eta_1} S, \end{aligned} \tag{5.71}$$

given the initial data, $(h_2, \vec{q}_2)_{i,j}^{n+1*}$. Where $\vec{\boldsymbol{u}}_{\eta_1} = (u(\vec{\boldsymbol{X}}, t), v(\vec{\boldsymbol{X}}, t))^T\Big|_{z=\eta_1}$. With the intermediate solutions, $(A_{1i}, Q_{1i})^{n+1*}$ and $(h_2, \vec{q}_2)_{i,j}^{n+1*}$, then using forward Euler time discretization, we approximate the solution of (5.70) and (5.71) as

$$\begin{aligned} A_{1,i}^{n+1} &= A_{1,i}^{n+1*} - S_i^A \Delta t, \\ Q_{1,i}^{n+1} &= Q_{1,i}^{n+1*} - S_i^Q \Delta t, \end{aligned} \tag{5.72}$$

and

$$\begin{aligned} h_{2,i,j}^{n+1} &= h_{2,i,j}^{n+1*} + S_{i,j} \Delta t, \\ \vec{q}_{2,i,j}^{n+1} &= \vec{q}_{2,i,j}^{n+1*} + \vec{\boldsymbol{u}}_{\eta_1,i,j} S_{i,j} \Delta t, \end{aligned} \tag{5.73}$$

respectively, where

$$S_{i,j} \approx S|_{T_{i,j}}, \quad \vec{\boldsymbol{u}}_{\eta_1,i,j} \approx \vec{\boldsymbol{u}}_{\eta_1}\Big|_{T_{ij}}, \quad S_i^A \approx \int_{K_i} S dy \quad \text{and } S_i^Q \approx \int_{K_i} u_{\eta_1} S dy.$$

To complete the description of the scheme, we need to define the exchange terms and interface velocities; $S_{i,j}, \vec{u}_{\eta_1,i,j}, S_i^A$ and S_i^Q. But there are no equations for these

terms. However they can be determined by respecting the following conditions :
(i) Conservation of mass and momentum.
(ii) Discrete version of (5.32).
(iii) Discrete consistency requirement (definition 5.6.1) of the resulting solution and
(iv) And non-negativity of water heights.
The above conditions allow to first obtain the heights, hence the exchange terms and interface velocities are calculated.

Lower layer Area

For global conservation, the total mass/momentum leaving one layer must equal the mass/momentum entering the other layer. Hence we assume the following :

$$S_i^A = \sum_j S_{i,j}\Delta y_{ij} \text{ and } S_i^Q = \sum_j u_{\eta_1,i,j} S_{i,j}\Delta y_{ij}. \tag{5.74}$$

Recall

$$A_i^{n+1} := A_{1,i}^{n+1} + \sum_j h_{2,i,j}^{n+1}\Delta y_j. \tag{5.75}$$

Substituting the expressions for $A_{1,i}^{n+1}$ and $h_{2,i,j}^{n+1}$ in (5.72) and (5.73) into (5.75), making use of the first conservation property, (5.74), then (5.75) gives

$$A_i^{n+1} = A_{1,i}^{n+1*} + \sum_j h_{2,i,j}^{n+1*}\Delta y_{ij}. \tag{5.76}$$

This explicitly gives the total wetted area, A_i^{n+1} directly from the intermediate solutions (n+1* values). Next, the discrete version of (5.32) is

$$A_{1,i}^{n+1} = \min\left(A_i^{n+1}, A_{c,i}\right). \tag{5.77}$$

Upper layer Heights

To compute the upper layer heights, we use the definition of $A_{1,i}^{n+1}$ in (5.77) alongside the relations (5.75) and (5.76). Before we continue, lets define the following notations, $A_{2,i}^{n+1*} := \sum_j h_{2,i,j}^{n+1*}\Delta y_{ij}$ and $A_{2,i}^{n+1} := \sum_j h_{2,i,j}^{n+1}\Delta y_{ij}$. We consider the following two cases based on the solution of $A_{1,i}^{n+1}$ in (5.77).

Case 1 : If $A_{1,i}^{n+1} = A_i^{n+1}$ (Lower Layer at t^{n+1} not full)

Then by (5.75), we have $\sum_j h_{2,i,j}^{n+1} \Delta y_{ij} = A_i^{n+1} - A_{i,i}^{n+1} = 0$. Since $h_{2,i,j}^{n+1} \geq 0$, hence

$$h_{2,i,j}^{n+1} = 0. \tag{5.78}$$

Case 2 : If $A_{1,i}^{n+1} = A_{c,i}$ (Lower Layer at t^{n+1} full)

Then, subtracting (5.76) from (5.75) gives

$$A_{2,i}^{n+1} = A_{2,i}^{n+1*} + \left(A_{1,i}^{n+1*} - A_{c,i} \right) \tag{5.79}$$

and consider two further cases.

Case 2a : If $A_{1,i}^{n+1*} - A_{c,i} \geq 0$ (Lower Layer full at intermediate state, t^{n+1*})

Then $A_{2,i}^{n+1} \geq A_{2,i}^{n+1*}$ by an amount $A_{excess,i}^{n+1*} := A_{1,i}^{n+1*} - A_{c,i} \geq 0$, so we add the constant excess height, $h_{excess,i}^{n+1*} = A_{excess,i}^{n+1*}/B_i$ to the intermediate solution $h_{2,i,j}^{n+1*}$, namely

$$h_{2,i,j}^{n+1} = h_{2,i,j}^{n+1*} + \frac{A_{excess,i}^{n}}{B_i}. \tag{5.80}$$

Case 2b : If $A_{1,i}^{n+1*} - A_{c,i} < 0$ (Lower Layer not full at intermediate state, t^{n+1*})

Then $A_{2,i}^{n+1} < A_{2,i}^{n+1*}$ by the amount, $A_{gap,i}^{n+1*} = A_{c,i} - A_{1,i}^{n+1*} < 0$. Hence we remove $A_{gap,i}^{n+1*}$ from $A_{2,i}^{n+1*}$ using the following algorithm.

- initialize $h_{2,i,j}^{n+1} = h_{2,i,j}^{n+1*}$ for all T_{ij}.
- $h_{gap,i}^{n+1*} = \frac{A_{gap,i}^{n+1*}}{B_i}, \quad TOL = 10^{-12}$.
- while($h_{gap,i}^{n+1*} > TOL$)

  i for all T_{ij}

  (a) $h_t = h_{2,i,j}^{n+1}$.

  (b) Reduce upper layer height by the gap height :

$$h_{2,i,j}^{n+1} = max(0, h_{2,i,j}^{n+1} - h_{gap,i}^{n+1*}). \tag{5.81}$$

  (c) Remove area of reduced height from total gap area:
  $A_{gap,i}^{n+1*} = A_{gap,i}^{n+1*} - |h_{2,i,j}^{n+1} - h_t| \Delta y_{i,j}$.

ii $h_{gap,i}^{n+1*} = \frac{A_{gap,i}^{n+1*}}{B_i}$.

Equations (5.78), (5.80) and (5.81) compute $h_{2,i,j}^{n+1}$ in $T_{ij} \in K_i$ for all three cases above.

Computing the Exchange Terms and the Interface velocity

Having computed $h_{2,i,j}^{n+1}$, we now compute the exchange terms, $S_{i,j}$ using the first equation in (5.73), hence

$$S_{i,j} = \frac{h_{2,i,j}^{n+1} - h_{2,i,j}^{n+1*}}{\Delta t}. \tag{5.82}$$

Next, we compute the interface velocity, $\vec{u}_{\eta_1,i,j}$. As proposed for multilayer systems in [Audusse et al., 2011], if water moves from lower layer to upper layer ($S_{i,j} > 0$), then the interface velocity, $\vec{\boldsymbol{u}}_{\eta_1,i,j}$ is that of the lower layer. But if the reverse is the case ($S_{i,j} \leq 0$), then $\vec{\boldsymbol{u}}_{\eta_1,i,j}$ is that of the upper layer. Hence we define

$$\vec{\boldsymbol{u}}_{\eta_1,i,j} := \begin{cases} \vec{u}_{2,i,j}^{n+1*} = \frac{\vec{q}_{2y,i,j}^{n+1*}}{h_{2,i,j}^{n+1*}}, & \text{if } S_{i,j} \leq 0 \\ (u_{1,i}^{n+1*}, 0)^T = \left(\frac{Q_{1,i}^{n+1*}}{A_{1,i}^{n+1*}}, 0\right)^T, & \text{if } S_{i,j} > 0. \end{cases} \tag{5.83}$$

Computing the Discharges

With the above definitions for the exchange terms and the interface velocity, then the upper layer discharge, $\vec{\boldsymbol{q}}_{2,i,j}^{n+1}$ is computed in (5.73), namely

$$\vec{q}_{2,i,j}^{n+1} = \vec{q}_{2,i,j}^{n+1*} + \vec{\boldsymbol{u}}_{\eta_1,i,j} S_{i,j} \Delta t. \tag{5.84}$$

As stated above, to conserve momentum the lower layer exchange term, S_i^Q is defined as

$$S_i^Q = \sum_j^{N_y^1} u_{\eta_1,i,j} S_{i,j} \Delta y_{ij}, \quad \text{see (5.74).} \tag{5.85}$$

Hence, the lower layer discharge is computed as

$$Q_{1,i}^{n+1} = Q_{1,i}^{n+1*} - S_i^Q \Delta t, \quad \text{see (5.72).} \tag{5.86}$$

This ensures that momentum is conserved. Hence, the discharges for both layers are obtained. Therefore, we now have $(A_{1,i}, Q_{1,i})^{n+1}$, and $(h_2, \vec{q}_2)_{i,j}^{n+1}$. With these

results, we update the full $2D$ data, $(H_{i,j}, \vec{q}_{i,j})^{n+1}$ in the next section which is the third and last step.

5.6.5 Step 3: Updating full $2D$ data $H_{i,j}^{n+1}$, $q_{x,i,j}^{n+1}$ and $q_{y,i,j}^{n+1}$ for $T_{i,j} \in K_i$

The full water depth, $H_{i,j}^{n+1}$ is given by

$$H_{i,j}^{n+1} = h_{1,i,j}^{n+1} + h_{2,i,j}^{n+1}, \tag{5.87}$$

where

$$h_{1,i,j}^{n+1} = \mathcal{H}(A_{1,i}^{n+1}, z_{bij}). \tag{5.88}$$

The full 2D x-discharge is computed as

$$\begin{aligned} q_{x,i,j}^{n+1} &:= q_{1x,i,j}^{n+1} + q_{2x,i,j}^{n+1} \\ &= h_{1,i,j}^{n+1} \frac{Q_{1,i}^{n+1}}{A_{1,i}^{n+1}} + q_{2x,i,j}^{n+1}. \end{aligned} \tag{5.89}$$

Finally, the lateral discharge is computed as follows :

$$q_{y,i,j}^{n+1} = H_{i,j}^{n+1} v_{i,j}^{n+1}, \tag{5.90}$$

where

$$v_{i,j}^{n+1} := v_{2,i,j}^{n+1} = \begin{cases} \frac{q_{2y,i,j}^{n+1*} + v_{\eta_1,i,j} S_{i,j} \Delta t}{h_{2,i,j}^{n+1}}, & \text{if } h_{2,i,j}^{n+1} > 0 \\ 0, & \text{else.} \end{cases} \tag{5.91}$$

This completes the numerical algorithm for the channel flow. As mentioned in the introductory section, the scheme for the floodplain flow is the 2D solver presented in section 3.4. At the channel/floodplain interface, the lateral fluxes are effortlessly computed since the channel 2D cell, $T_{ij} \in K_i \in \Omega_h^1$ has the complete 2D data, $(H_{i,j}, \vec{q}_{i,j})^n$. This is not the case for most existing methods since the lateral discharge is not easily available. In the VCM, the fluxes between the floodplains and channel are taken into account in the first stage of operator splitting step where the intermediate solutions are computed. Figures 5.8 and 5.9 are the flow charts for the implementation of the VCM.

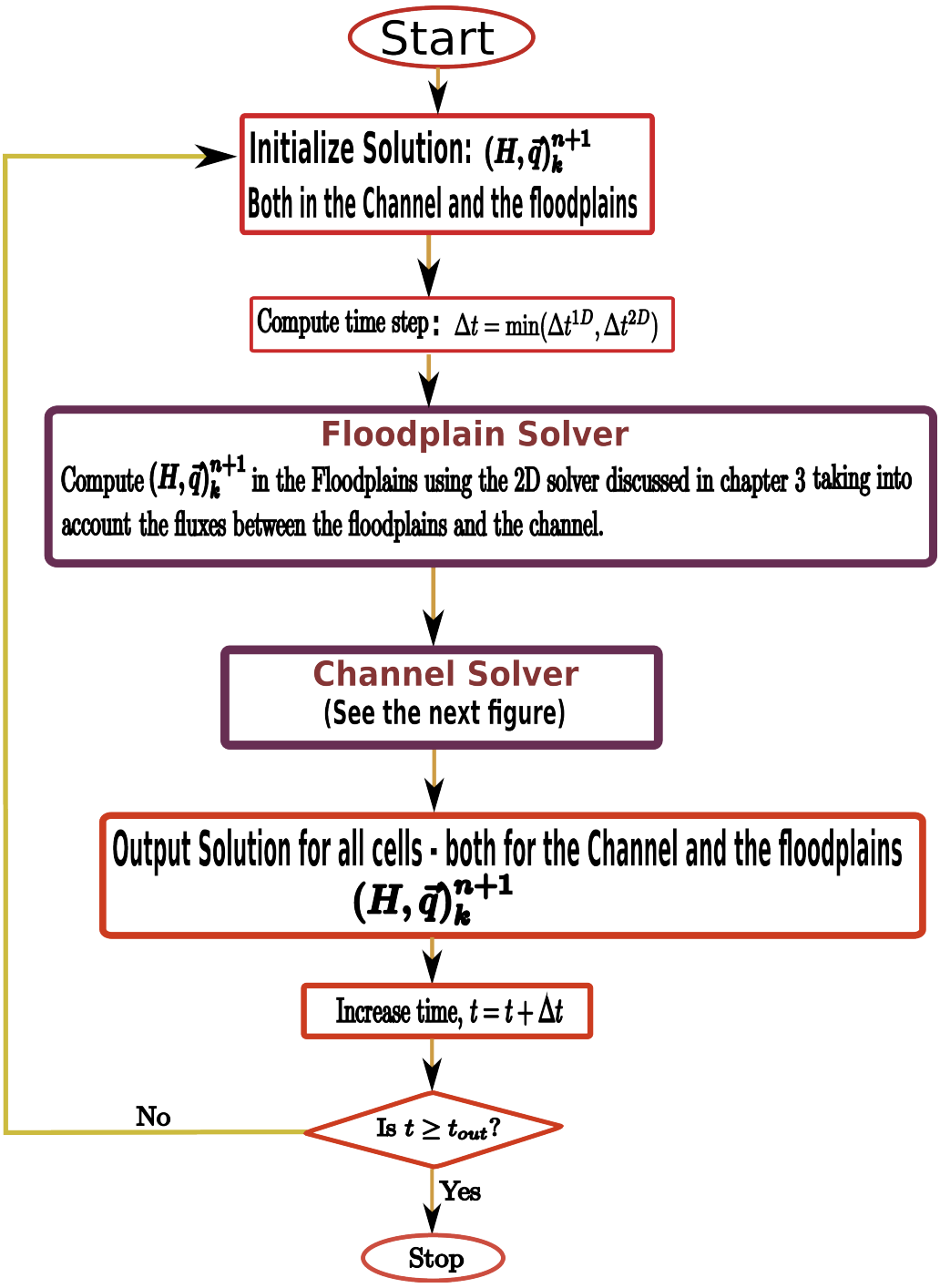


Figure 5.8: Flow chart for the VCM depicting the floodplain flow solver and the channel flow solver. The details of the channel solver is illustrated in figure 5.9.

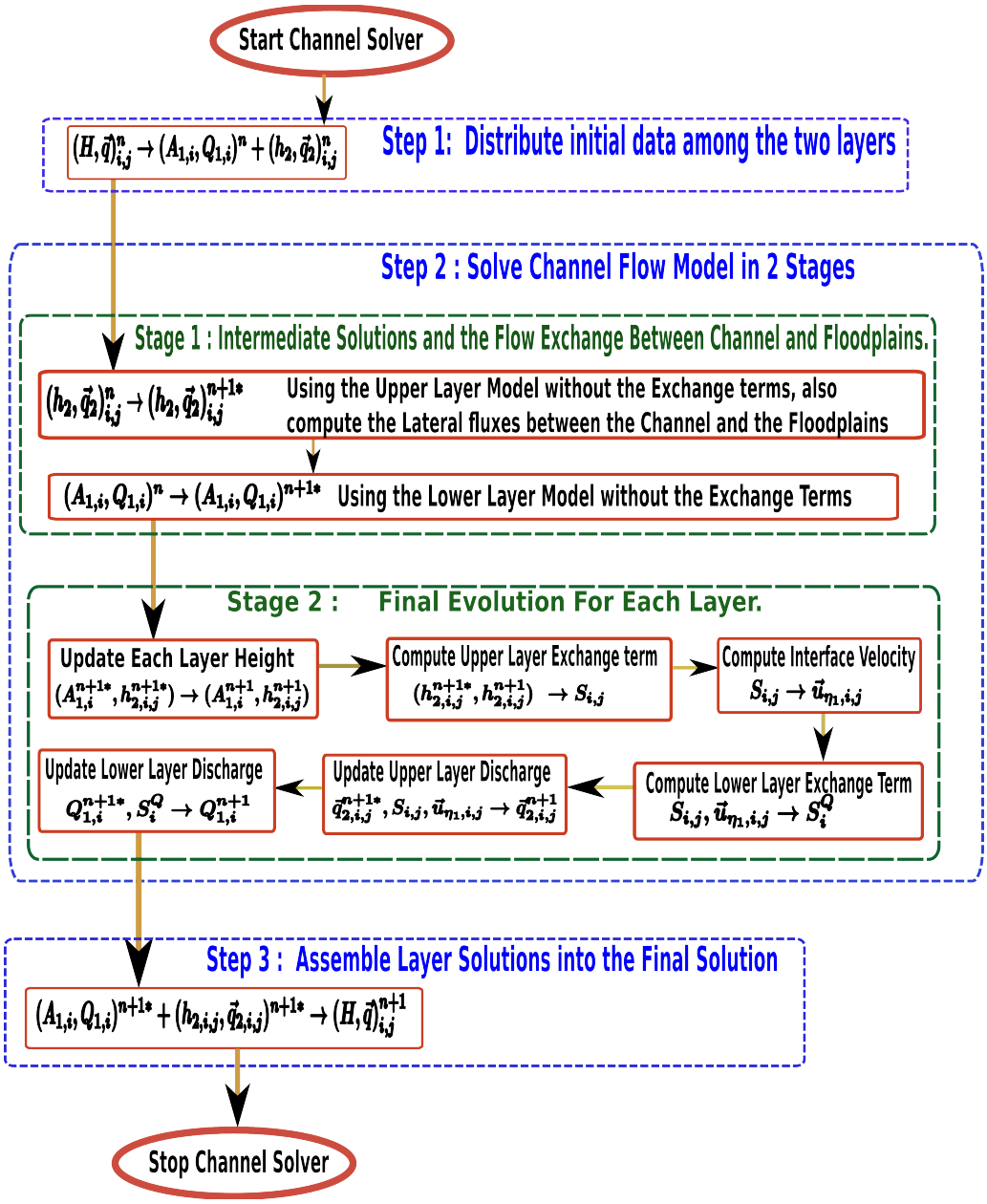


Figure 5.9: Details of the channel solver part in the flow chart of the VCM (figure 5.8). It depicts the three-step channel flow solver which implements the two-layer coupled models, (5.46), (5.47). It also shows the two stages to solve (step 2) the channel models. Step 2, stage 1 computes the intermediate solutions and the lateral fluxes between the channel and the floodplains, while step 2, stage 2 enforces the mass and momentum conservation to update each layer solution.

5.6.6 Finite Volume Methods for the intermediate solutions

The remaining task is to present the solution methods for the models, (5.68) and (5.69). Let us emphasis that the choice of method of solution for these models, is open to any appropriate PDE solver so long as it satisfies the properties for the solutions of hyperbolic conservation laws with source terms as mentioned in a previous chapter. Therefore, any of the finite volume methods or DG-FEM methods can be applied to these models. In the following sections, we adapt existing finite volume methods to approximate the solution of these sub-layer models and show that each method satisfies the well-balance of lake at rest.

5.6.7 Solution of the Upper Layer Model without Exchange Terms

As can be seen in the $2D$ model, (5.69) it can be interpreted as a $2D$ shallow water equation with topography, $z_b(\vec{\boldsymbol{X}}) + h_1(\vec{\boldsymbol{X}}, t)$ referred to as the apparent topography in [Bouchut, 2004, 2007]. Therefore, we propose the following apparent topography hydrostatic reconstruction scheme [Audusse et al., 2004; Bouchut, 2004, 2007]:

$$U^{n+1*}_{2,i,j} = U^{n}_{2,i,j} - \frac{\Delta t}{|T_{ij}|} \sum_{e_k \in \mathcal{E}_{ij}} |e_k| \Big(\phi^{2D}(\tilde{U}^{n}_{2,k,L}, \tilde{U}^{n}_{2,k,R}, \vec{n}_k) + s(h^{n}_{2,i,j}, \tilde{h}^{n}_{2,k,L}, \vec{n}_k) \Big) \tag{5.92}$$

where e_k is $k-$th edge of T_{ij}, with size, $|e_k|$ and outward unit normal, $\vec{n}_k$. L and R indicate the cells at left and right of the edge, $\mathcal{E}_{ij}$ is the set of all edges of T_{ij} and $U^{n+1*}_{2,i,j} = (h^{n+1*}_{2,i,j}, \vec{q}^{\,n+1*}_{2,i,j})^T$ is the upper layer ***intermediate solution*** which is being computed. And

$$\begin{aligned}
&\phi^{2D}(\tilde{U}^{n}_{2,k,L}, \tilde{U}^{n}_{2,k,R}, \vec{n}_k) = T^{-1}_{\vec{n}_k} \phi(T_{\vec{n}_k} \tilde{U}^{n}_{2,k,L}, T_{\vec{n}_k} \tilde{U}^{n}_{2,k,L}),\\
&s(h^{n}_{2,i,j}, \tilde{h}^{n}_{2,k,L}, \vec{n}_k) = \frac{g}{2} \Big(0, ([h^{n}_{2,i,j}]^2 - [\tilde{h}^{n}_{2,k,L}]^2) \vec{n}_k \Big)^T,\\
&\tilde{U}^{n}_{2,k,p} = \frac{\tilde{h}^{n}_{2,k,p}}{h^{n}_{2,k,p}} U^{n}_{2,k,p}.\\
&U^{n}_{2,i,j} = (h^{n}_{2,i,j}, \vec{q}^{\,n}_{2,i,j})^T.\\
&U^{n}_{2,k,p} = (h^{n}_{2,k,p}, \vec{q}^{\,n}_{2,k,p})^T.\\
&\tilde{h}^{n}_{2,k,p} = \max(\eta^{n}_{k,p} - \eta^{n*}_{1,k}, 0).\\
&\eta^{n}_{k,p} = \eta^{n}_{1,k,p} + h^{n}_{2,k,p}.\\
&p = L, R.
\end{aligned} \tag{5.93}$$

k is the edge number.

$$\begin{aligned} &\eta^n_{1,k,p} = z_{b,k,p} + h^n_{1,k,p} \quad \text{(The apparent topography on side } p = L, R), \\ &\eta^{n*}_{1,k} = \max(\eta^n_{1,k,R}, \eta^n_{1,k,L}). \end{aligned} \tag{5.94}$$

The quantities, $z_{b,k,p}, h_{1,k,p}, \eta^n_{1,k,p}$ and $\eta^n_{k,p}$ are respectively, the bottom elevation, water depth of lower layer, top height of lower layer, and free-surface elevation of 2D cell on side $P, (P = L, R)$ of the $k - th$ edge of cell, T_{ij}. ϕ is any consistent numerical flux function and $T_{\vec{n}_k}$ is the matrix of rotation as discussed in chapter 3.

5.6.8 Solution of the Lower Layer Model without Exchange terms

Recall that if the channel is full, then the lower layer top elevation ($z = \eta_1(x,t) = z^w_b(x)$) is different from the free-surface elevation, ($z = \bar{\eta}(x,t)$) appearing in the lower layer 1D model. In this case, the top of the lower layer will not generally be constant (in x) even when the free-surface is constant (in x). This leads to difficulty in deriving a well-balance method for the lower layer model. In order to tackle this problem, we propose to approximate the solution of the lower layer model in the following way. Consider the two systems of equations

$$\partial_t \begin{pmatrix} A \\ Q \end{pmatrix} = -\partial_x \begin{pmatrix} Q \\ \frac{Q^2}{A} \end{pmatrix} - \begin{pmatrix} 0 \\ gA\partial_x \bar{\eta} \end{pmatrix} + \Phi(x,t) \tag{5.95}$$

and

$$\partial_t \begin{pmatrix} A_2 \\ Q_2 \end{pmatrix} = -\partial_x \begin{pmatrix} Q_2 \\ \frac{Q_2^2}{A_2} \end{pmatrix} - \begin{pmatrix} 0 \\ gA_2\partial_x \bar{\eta} \end{pmatrix} + \Phi(x,t). \tag{5.96}$$

The system, (5.95) is the 1D model for the full channel flow as derived in chapter 4 and $\Phi(x,t) = (\Phi^A(x,t), \Phi^Q(x,t))^T$. The system, (5.96) is a 1D model for the upper layer obtained by integrating the upper layer 2D model and considering a laterally constant free surface. Substracting (5.96) from (5.95) gives

$$\partial_t \begin{pmatrix} A - A_2 \\ Q - Q_2 \end{pmatrix} = -\partial_x \begin{pmatrix} Q - Q_2 \\ \frac{(Q-Q_2)^2}{A-A_2} \end{pmatrix} - \begin{pmatrix} 0 \\ g(A - A_2)\partial_x \bar{\eta} \end{pmatrix} + \partial_x \begin{pmatrix} 0 \\ \frac{AA_2}{A-A_2}(\underline{u} - \underline{u}_2)^2 \end{pmatrix},$$

where $\underline{u} = \frac{Q}{A}$ and $\underline{u}_2 = \frac{Q_2}{A_2}$. Using the fact that the vertical variation of velocity is negligible, hence $(\underline{u} - \underline{u}_2)^2 \simeq 0$, we have

$$\partial_t \begin{pmatrix} A_1 \\ Q_1 \end{pmatrix} \simeq \partial_t \begin{pmatrix} A - A_2 \\ Q - Q_2 \end{pmatrix} \tag{5.97}$$

since

$$\partial_t \begin{pmatrix} A_1 \\ Q_1 \end{pmatrix} = -\partial_x \begin{pmatrix} Q_1 \\ \frac{Q_1^2}{A_1} \end{pmatrix} - \begin{pmatrix} 0 \\ gA_1\partial_x\bar{\eta} \end{pmatrix}.$$

Hence, from (5.97), we propose to approximate the solution of the lower layer model, (5.68) as

$$\begin{pmatrix} A_{1,i} \\ Q_{1,i} \end{pmatrix}^{n+1*} = \begin{pmatrix} A_i \\ Q_i \end{pmatrix}^{n+1*} - \begin{pmatrix} A_{2,i} \\ Q_{2,i} \end{pmatrix}^{n+1*}. \tag{5.98}$$

where $(A_i^{n+1*}, Q_i^{n+1*})^T$ is the approximate solution of the full layer 1D model, (5.95) solved with any known method and $(A_{2,i}^{n+1*}, Q_{2,i}^{n+1*})^T \simeq \sum_j \left(h_{2,i,j}^{n+1*}, q_{2x,i,j}^{n+1*}\right)^T \Delta y_{i,j}$ is obtained from the $2D$ solution, (5.92). So we do not need to solve (5.96), rather we approximate its solution with those of the 2D upper layer. In this thesis, we use the scheme of [Morales-Hernández et al., 2013], presented in a previous chapter, for the solution of the model, (5.95).

In summary, what we have proposed for the lower layer model, is to solve (5.95), use the solution of (5.92), then apply (5.98) to obtain the lower layer intermediate solution. It is important to reiterate that this approach is just one possible way of solving the lower layer model, (5.68). There might be other ways of solving this model and still achieve the desired properties. We prove in theorems 5.7.6 and 5.7.7 that if the underlying solver for (5.95) is well-balanced and mass conservative, then the lower layer scheme, (5.98) is also well-balanced and mass conservative.

5.7 Desirable Properties of the VCM Scheme

In order to evolve the solution from time step t^n to t^{n+1}, we require the following conditions to be satisfied.

Definition 5.7.1 (Consistency of Distribution Operation)**.** *We require a distribution operation (step 1) such as the one in section 5.6.3 to be* ***consistent*** *in the sense that whenever the channel is not full ($H_{i,j}^n \leq \beta_{ij} \quad \forall T_{ij} \in K_i$), then the following*

conditions must hold:

i $h^n_{2i,j}, q^n_{2xi,j}$ *and* $q^n_{2yi,j}$ *each equal zero* $\forall T_{ij} \in K_i$;

ii The lower layer area, $A^n_{1,i}$ *must correspond to total water in cross-section. That is,* $A^n_{1i} = \mathcal{A}(\eta^n_{i,1}, \eta^n_{i,2}, ..., \eta^n_{i,N^1_y})$ *(*$= \sum_j H^n_{i,j}\Delta y_j$ *if channel is locally rectangular).*

These ensure that we do not solve two-layer problem when the flow is actually one-layer (not full) and in such case, the lower layer must account for the total water in the cross-section. This is the self-adaptive nature of the scheme.

Definition 5.7.2 (No-Numerical Flooding Property)**.** *In order to satisfy the no-numerical flooding property, the solution of the scheme,* $(H_{i,j}, q_{xi,j}, q_{yi,j})^{n+1}$ *must satisfy the following conditions:*

i

$$(\beta_{ij} - h^{n+1}_{1i,j})h^{n+1}_{2i,j} = 0 \quad \forall T_{ij} \in K_i; \tag{5.99}$$

ii If $h^{n+1}_{2,i,j} = 0$*, then (a)* $q^{n+1}_{2x,i,j} = 0$ *(b)* $u^{n+1}_{i,j} = \frac{Q^{n+1}_{1,i}}{A^{n+1}_{1,i}}$ *(which is constant laterally) and (c)* $q^{n+1}_{y,i,j} = 0$*;*

iii If $\exists j^* \in \{1, 2, ..., N^1_y\} \mid \eta^{n+1}_{i,j^*} < \eta^{\beta}_i$*, then* $\eta^{n+1}_{i,j} = \eta^{n+1}_{i,j^*} \forall j$*.*
This ensures that the solution at $t^n + 1$ *satisfies the discrete consistency requirement (see definition 5.6.1).*

Condition (i) means that lower layer is either full ($h_{1i,j} = \beta_{i,j}$) or upper layer is dry $h_{2i,j} = 0$. In other words, there shouldn't be gap between the two layers. This way, water would not overflow when the channel is not yet full. Condition (ii) states that if upper layer is dry, then the flow must be $1D$. That is, if upper layer is dry, then its velocities and discharges must be zero. And the full layer data must be laterally uniform. Condition (iii) means that if channel is not full, then the free-surface must be flat (equal) in lateral direction. This is the discrete consistency requirement (definition 5.6.1) on the solution at t^{n+1}.

In addition, we also prove that the scheme is well-balanced and conservative under suitable conditions on the intermediate solutions.

Theorem 5.7.1 (Consistency of Distribution Operation)**.** *The distribution operation proposed in section 5.6.3 is consistent with the problem in the sense of definition 5.7.1.*

Proof. We need to prove (i) and (ii) in definition 5.7.1.
i) If the channel is not full, then

$$H_{i,j}^n \leq \beta_{ij}.$$

Hence, by (5.67), we have

$$h_{2,i,j}^n = H_{i,j}^n - \min(H_{i,j}, \beta_{ij}) = H_{i,j}^n - H_{i,j}^n = 0.$$

Consequently,

$$q_{2x,i,j}^n = q_{2y,i,j}^n = 0 \text{ by (5.67).}$$

ii)

$$H_{i,j}^n \leq \beta_{i,j} => z_{bij} + H_{i,j}^n \leq z_{bij} + \beta_{i,j} => \eta_{i,j}^n \leq \eta_i^\beta.$$

Hence, by discrete consistency (definition 5.6.1) , we have

$$\bar{\eta}_i = \eta_{i,j}^n < \eta_i^\beta \forall j = 1, ..., N_y^1.$$

By (5.66),

$$\eta_{1,i}^n := \min(\bar{\eta}_i^n, \eta_i^\beta) = \bar{\eta}_i^n = \eta_{i,j}^n \forall j = 1, ..., N_y^1.$$

Hence,

$$A_{1,i}^n := \mathcal{A}(\eta_{1,i}^n, \eta_{1,i}^n, ..., \eta_{1,i}^n) = \mathcal{A}(\eta_{i,1}^n, \eta_{i,2}^n, ..., \eta_{i,N_y^1}^n) =: \text{ total wetted area.}$$

□

Theorem 5.7.2 (No-Numerical Flooding Property)**.** *The vertical coupling scheme as derived in* (5.87)*,* (5.89) *and* (5.90)*, preserves the no-numerical flooding property in the sense of definition 5.7.2.*

Proof. i We prove (5.99) on case-by-case bases.

Case 1:

$$h_{2,i,j}^{n+1} = 0, \quad \text{see (5.78).}$$

Cases 2a and 2b :

$$A_{1,i}^{n+1} = A_{c,i} => h_{1,i,j}^{n+1} := \mathcal{H}(A_{1,i}^{n+1}, z_{bij}) = \mathcal{H}(A_{c,i}, z_{bij}) = \beta_{ij}.$$

Therefore,

$h_{2,i,j}^{n+1} = 0$ or $h_{1,i,j}^{n+1} = \beta_{ij}$ in all cases. Hence,

$$(\beta_{ij} - h_{1i,j}^{n+1})h_{2i,j}^{n+1} = 0 \quad \forall T_{ij} \in K_i$$

as claimed.

ii Let $h_{2,i,j}^{n+1} = 0$, then
(a)

$$S_{i,j} = -\frac{h_{2,i,j}^{n+1*}}{\Delta t} < 0 => u_{\eta_1,i,j} = u_{2,i,j}^{n+1*} => u_{\eta_1,i,j} S_{i,j} \Delta t = q_{2x,i,j}^{n+1*} => q_{2x,i,j}^{n+1} = 0.$$

(b) Since $q_{2x,i,j}^{n+1} = 0$, then

$$q_{x,i,j}^{n+1} = h_{1,i,j}^{n+1} \frac{Q_{1,i}^{n+1}}{A_{1,i}^{n+1}} = \frac{Q_{1,i}^{n+1}}{A_{1,i}^{n+1}} H_{i,j}^{n+1} \text{ (because} h_{2,i,j}^{n+1} = 0).$$

Hence,

$$u_{i,j}^{n+1} := \frac{q_{x,i,j}^{n+1}}{H_{i,j}^{n+1}} = \frac{Q_{1,i}^{n+1}}{A_{1,i}^{n+1}} \text{ (which is constant in } j).$$

(c)

$$v_{i,j}^{n+1} = 0 \text{ (see(5.91))} => q_{y,i,j}^{n+1} = 0.$$

iii Let there exist $j^* \in \{1, 2, ..., N_y^1\}$ such that $\eta_{i,j*}^{n+1} < \eta_i^{\beta}$, then this corresponds to case 1 in step 2 because cases 2a and 2b satisfy

$$A_{1,i}^{n+1} = A_{c,i} => \eta_{i,j}^{n+1} \geq \eta_i^{\beta} \forall j = 1, ..., N_y^1.$$

Since, $\eta_{i,j*}^{n+1} < \eta_i^{\beta}$ corresponds to case 1, then it satisfies

$$A_{1,i}^{n+1} = A_i^{n+1} \leq A_{c,i} => h_{2,ij}^{n+1} = 0 \forall j.$$

So,

$$\eta_{i,j}^{n+1} := H_{i,j}^{n+1} + z_{bij} = h_{1,ij}^{n+1} + z_{bij} = \eta_{1,i}^{n+1} \forall j = 1, ..., N_y^1.$$

That is,

$$\eta_{i,j}^{n+1} = \eta_{1,i}^{n+1} = \eta_{i,j^*}^{n+1} \forall j = 1, .., N_y^1.$$

Hence the free surface is laterally flat. Therefore, the discrete consistency requirement (definition 5.6.1) holds at t^{n+1}. Therefore, the scheme satisfies the no-numerical flooding property.

□

Theorem 5.7.3 (Well-balance)**.** *If the numerical schemes for the intermediate solutions,* $(A_{1,i}^{n+1*}, Q_{1,i}^{n+1*})^T$ *and* $(h_{2,i,j}^{n+1*}, q_{2x,i,j}^{n+1*}, q_{2y,i,j}^{n+1*})^T$ *are well-balanced, then the vertical coupling method,* (5.87)*,* (5.89) *and* (5.90)*, is well-balanced. This is true for any kind of well-balance, not only for lake at rest.*

Proof. Let the intermediate solutions be well-balanced, then

$$(A_{1,i}^{n+1*}, Q_{1,i}^{n+1*})^T = (A_{1,i}^{n}, Q_{1,i}^{n})^T,$$
$$(h_{2,i,j}^{n+1*}, q_{2x,i,j}^{n+1*}, q_{2y,i,j}^{n+1*})^T = (h_{2,i,j}^{n}, q_{2x,i,j}^{n}, q_{2y,i,j}^{n})^T.$$

Recall that

$$\begin{aligned} A_{1,i}^{n} &= \min(A_i^n, A_{c,i}), \\ A_{1,i}^{n+1} &:= \min(A_i^{n+1}, A_{c,i}) = \min(A_{1,i}^{n+1*} + A_{2,i}^{n+1*}, A_{c,i}) \\ &= \min(A_{1,i}^{n} + A_{2,i}^{n}, A_{c,i}) \text{ (intermediate solutions are well-balanced)} \\ &= A_{1,i}^{n}. \end{aligned}$$

Hence, $h_{1,i,j}^{n+1} = h_{1,i,j}^{n}$.

Next, we show that $h_{2,i,j}^{n+1} = h_{2,i,j}^{n}$.

$$\begin{aligned} A_{2,i}^{n+1} &:= A_i^{n+1} - A_{1,i}^{n+1} = (A_{1,i}^{n+1*} + A_{2,i}^{n+1*}) - A_{1,i}^{n} \\ &= (A_{1,i}^{n} + A_{2,i}^{n+1*}) - A_{1,i}^{n} = A_{2,i}^{n+1*} \\ &= A_{2,i}^{n} => A_i^{n+1} = A_i^n. \end{aligned}$$

Case 1: $A_{1,i}^{n+1} = A_i^{n+1} < A_{c,i}$, that is

$$A_{1,i}^{n+1} = A_i^{n+1} = A_i^n < A_{c,i}.$$

But $A_i^n < A_{c,i} => h_{2,i,j}^{n} = 0$, that is

$$h_{2,i,j}^{n+1*} = h_{2,i,j}^{n} = 0.$$

Also,

$$A_i^{n+1} < A_{c,i} => h_{2,i,j}^{n+1} = 0 = h_{2,i,j}^{n}.$$

Cases 2a and 2b: $A_{1,i}^{n+1} = A_{c,i}$, that is

$$A_{1,i}^{n+1} = A_{1,i}^{n} = A_{1,i}^{n+1*} = A_{c,i}.$$

But

$$A_{1,i}^{n+1*} = A_{c,i} => A_{excess,i}^{n+1*} = A_{gap}^{n+1*} = 0 => h_{excess,i}^{n+1*} = h_{gap}^{n+1*} = 0.$$

Hence by (5.80) and (5.81), we have

$$h_{2,i,j}^{n+1} = h_{2,i,j}^{n+1*} = h_{2,i,j}^{n}.$$

Finally, we show that $(H, \vec{q}_2)_{i,j}^{n+1} = (H, \vec{q}_2)_{i,j}^{n+1}$.

$$\begin{aligned}
&=> H_{i,j}^{n+1} = h_{1,i,j}^{n} + h_{2,i,j}^{n} = H_{i,j}^{n} \text{ and } S_{i,j} = 0 \\
&=> q_{2x,i,j}^{n+1} = q_{2x,i,j}^{n+1*} = q_{2x,i,j}^{n}, \quad q_{2y,i,j}^{n+1} = q_{2y,i,j}^{n+1*} = q_{2y,i,j}^{n}, \text{ and } Q_{i,1}^{n+1} = Q_{1,i}^{n+1*} = Q_{1,i}^{n} \\
&=> q_{x,i,j} := h_{1,i,j}^{n+1} \frac{Q_{1,i}^{n+1}}{A_{1,i}^{n+1}} + q_{2y,i,j}^{n+1} = h_{1,i,j}^{n} \frac{Q_{1,i}^{n}}{A_{1,i}^{n}} + q_{2y,i,j}^{n} = q_{x,i,j}^{n} \\
&\text{and } q_{y,i,j}^{n+1} := H_{i,j}^{n+1} \frac{q_{2y,i,j}^{n+1}}{h_{2,i,j}^{n+1}} = H_{i,j}^{n} \frac{q_{2y,i,j}^{n}}{h_{2,i,j}^{n}} = q_{y,i,j}^{n}.
\end{aligned}$$

We have shown that $H_{i,j}^{n+1} = H_{i,j}^{n}, q_{x,i,j}^{n+1} = q_{x,i,j}^{n}$ and $q_{y,i,j}^{n+1} = q_{y,i,j}^{n}$, so the method is well-balanced.

□

Theorem 5.7.4 (Conservation)**.** *If the intermediate solutions is mass conservative, then the vertical coupling solutions is also mass conservative.*

Proof. Let the intermediate solutions be mass conservative, then

$$\sum_i A_{1,i}^{n+1*} = \sum_i A_{1,i}^{n} \text{ and } \sum_i A_{2,i}^{n+1*} = \sum_i A_{2,i}^{n}. \tag{5.100}$$

Hence

$$\begin{aligned}
\sum_i (A_{1,i}^{n+1} + A_{2,i}^{n+1}) &:= \sum_i (A_{1,i}^{n+1*} + A_{2,i}^{n+1*}) \quad \text{(see (5.76))} \\
&= \sum_i (A_{1,i}^{n} + A_{2,i}^{n}) \quad \text{(see (5.100)).}
\end{aligned}$$

□

In the following, we prove that the proposed schemes for the intermediate solutions satisfy the desirable properties. Since the hydrostatic reconstruction method is mass conservative, [Audusse and Bristeau, 2005; Audusse et al., 2004] and the $2D$ models, (5.69) do not introduce any source term to the height equation, then the scheme (5.92) is mass conservative. We state and prove a theorem below to show

that this scheme, like the standard hydrostatic reconstruction [Audusse et al., 2004] method, preserves well-balance of lake at rest.

Theorem 5.7.5. *The upper layer scheme in* (5.92) *is well-balanced with respect to lake at rest.*

Proof. Let the flow satisfy the late at rest condition, then $\eta^n_{k,L} = \eta^n_{k,R}$ and $\vec{q}^n_{2,i,j} = 0$. Hence by (5.93), we have

$$\begin{aligned}\tilde{h}^n_{2,k,L} &:= \max(\eta^n_{k,L} - \eta^{n*}_{1,k}, 0) \\ &= \max(\eta^n_{k,R} - \eta^{n*}_{1,k}, 0) \quad \text{(because } \eta^n_{k,L} = \eta^n_{k,R}\text{)} \\ &= \tilde{h}^n_{2,k,R} \quad \text{(by definition).}\end{aligned} \tag{5.101}$$

Hence

$$\begin{aligned}\tilde{U}^n_{2,k,R} &:= \frac{\tilde{h}^n_{2,k,R}}{h^n_{2,k,R}} U^n_{2,k,R} = (\tilde{h}^n_{2,k,R}, 0, 0)^T \\ &= (\tilde{h}^n_{2,k,L}, 0, 0)^T \quad \text{(see (5.101))} \\ &= \tilde{U}^n_{2,k,L}.\end{aligned}$$

Therefore, by the consistency of numerical flux function, we have

$$\phi^{2D}(\tilde{U}^n_{2,k,L}, \tilde{U}^n_{2,k,R}, \vec{n}_k) = \begin{pmatrix} 0 \\ \frac{g}{2}(\tilde{h}^n_{2,k,L})^2 \vec{n}_k \end{pmatrix}.$$

Hence the scheme, (5.92) becomes

$$\begin{aligned}U^{n+1*}_{2,i,j} &= U^n_{2,i,j} - \frac{\Delta t}{|T_{ij}|} \sum_{e_k \in \mathcal{E}_{ij}} |e_k| \left[(0, \frac{g}{2}(\tilde{h}^n_{2,k,L})^2 \vec{n}_k)^T + \frac{g}{2} \left(0, ([h^n_{2,i,j}]^2 - [\tilde{h}^n_{2,k,L}]^2) \vec{n}_k \right)^T \right] \\ &= U^n_{2,i,j} - \frac{\Delta t}{|T_{ij}|} \sum_{e_k \in \mathcal{E}_{ij}} |e_k| \left[\frac{g}{2} \left(0, (h^n_{2,i,j})^2 \vec{n}_k \right)^T \right] \\ &= U^n_{2,i,j} - \frac{\Delta t}{|T_{ij}|} \frac{g}{2} (h^n_{2,i,j})^2 \sum_{e_k \in \mathcal{E}_{ij}} \left[\left(0, \vec{n}_k |e_k| \right)^T \right] = U^n_{2,i,j}.\end{aligned}$$

□

Theorem 5.7.6. *If the underlying solver for* (5.95) *is well-balanced then the lower layer scheme,* (5.98) *is also well-balanced.*

Proof. Let scheme for (5.95) be well-balanced, then $(A^{n+1*}_i, Q^{n+1*}_i)^T = (A^n_i, Q^n_i)^T$. Since the $2D$ solver, (5.92), is also well-balanced, so $(h^{n+1*}_{2,i,j}, \vec{q}^{n+1*}_{2,i,j})^T = (h^n_{2,i,j}, \vec{q}^n_{2,i,j})^T$.

Therefore, the lower layer scheme, (5.98) becomes

$$\begin{aligned}\begin{pmatrix} A_{1,i} \\ Q_{1,i} \end{pmatrix}^{n+1*} &:= \begin{pmatrix} A_i \\ Q_i \end{pmatrix}^{n+1*} - \begin{pmatrix} A_{2,i} \\ Q_{2,i} \end{pmatrix}^{n+1*} = \begin{pmatrix} A_i \\ Q_i \end{pmatrix}^{n} - \begin{pmatrix} A_{2,i} \\ Q_{2,i} \end{pmatrix}^{n} \\ &= \begin{pmatrix} A_{1,i} \\ Q_{1,i} \end{pmatrix}^{n}\end{aligned}$$

So, the $1D$ scheme is well-balanced. □

Theorem 5.7.7. *If the underlying solver for* (5.95) *is mass conservative, then the lower layer scheme,* (5.98) *is also mass-conservative.*

Proof. Let scheme for (5.95) be mass conservative, then $\sum_i A_i^{n+1*} = \sum_i A_i^n$. Since the $2D$ solver, (5.92), is also mass conservative, so $\sum_i A_{2,i}^{n+1*} = \sum_i A_{2,i}^n$. Therefore, the lower layer scheme, (5.98) gives

$$\begin{aligned}\sum_i A_{1,i}^{n+1*} &:= \sum_i A_i^{n+1*} - \sum_i A_{2,i}^{n+1*} \\ &= \sum_i A_i^n - \sum_i A_{2,i}^n \quad \text{(by conservation of } A_i^{n+1} \text{ and } A_{2,i}^{n+1*}\text{)} \\ &= \sum_i \left(A_i^n - A_{2,i}^n \right) = \sum_i A_{1,i}^n \quad \text{(by definition).}\end{aligned}$$

So, the $1D$uscheme is mass conservative. □

Since we use the $1D$ solver in [Morales-Hernández et al., 2013] which is conservative and well-balanced for lake at rest, and the $2D$ scheme, (5.92) is also mass conservative and well-balanced for lake at rest, therefore by theorems 5.7.4 and 5.7.3 the vertical coupling scheme, (5.87), (5.89), (5.90) is mass conservative and well-balanced for lake at rest.

This completes the theoretical aspect of the VCM. In the next chapter, we present some numerical experiments to access the its performance.

Chapter 6

Numerical Results for Coupling Methods

6.1 Introduction

In this chapter, we present the numerical results for the methods proposed in this thesis. All the algorithms described in this thesis are implemented in a self-written c++ code. The experiments are run on the Cluster of Workstations (COW) of the Center for Scientific Computing (CSC), University of Warwick. The CSC's COW is a heterogeneous cluster with approximately 600 cores for code development and small-scale computations.

In all the numerical experiments presented in this thesis, the floodplain flow model is solved with the 2D scheme described in chapter 3. Three test cases, including the one presented in chapters 3.8.3 and 4.8, are used in this chapter to access the performance of the methods. To discuss the results, we use displays of some or all the flow quantities in the whole computational domain and also plots of flow quantities at selected points. We access the methods for accuracy (with full 2D simulation as reference solution), efficiency and their abilities to recover 2D flow structure within the channel during flood. The acronym, FULL 2D refers to the solution obtained from solving the 2D shallow water equations in the entire flow domain (floodplain and channel). It is taken as the reference solution in each test case.

As stated in chapter 5, the VCM is a unifying method depending on the choice of the elevation function, $z_b^w(x)$. We wish to elaborate more on this. Recall from the formulation of the vertical coupling method, that $z_b^w(x)$ is the interface boundary between the lower and upper layers, if flooding (see figures 5.7(a) and 5.7(b) for the

discrete version, η_i^{β}). This means that changing the position of $z_b^w(x)$ would change the position of the interface and ultimately change the type of method. For instance,

- If
$$z_b^w(x) = Z_b(x) \text{ for all } x, \tag{6.1}$$
where $Z_b(x)$ is the bottom of the channel, then the vertical coupling method becomes a full 2D simulation because the lower layer no longer exists and upper layer model reduces to the full 2D shallow water equations.

- If
$$z_b^w(x) = \infty \text{ for all } x, \tag{6.2}$$
then the vertical coupling method reduces to the method proposed in chapter 4, which for the case where the channel walls are parallel to x-axis, reduces to the FBM as the lateral discharge would not be calculated in the channel. This is true because the upper layer vanishes and the solution of the lower layer model, as approximated in section 5.6.8, reduces to the channel model in chapter 4 but without the channel lateral discharge solver.

- Suppose it is known a priori that the channel never floods in some region, $\Omega_{nF} \in \mathbb{R}$ and there are other region, $\Omega_F \in \mathbb{R}$ in which the channel is most likely to flood, say where the channel flows into a floodplain, then we can define
$$z_b^w(x) = \begin{cases} 0, & \text{if } x \in \Omega_F, \\ \infty, & \text{if } x \in \Omega_{nF}, \end{cases} \tag{6.3}$$
and we ensure that $z_b^w(x)$ varies smoothly between Ω_F and Ω_{nF}. In this case, the vertical coupling method, is automatically a flow direction or frontal coupling method. Frontal or flow direction coupling method is the approach used to couple a 1D and 2D models when the 2D domain is in the direction of the 1D flow.

- If we define $z_b^w(x)$ to the channel wall elevation, as defined in chapter 5, then the method is a purely vertical coupling method, different from any existing method.

In this thesis, we do not consider all the cases but only focus on the last case

in which we take $z_b^w(x)$ as the channel wall elevation at point x. Before we continue, let us make the following remark.

Remark 6.1.1. *We observed that if $z_b^w(x)$ has very large gradients, then it leads to numerical difficulties for the vertical coupling method. Hence, we propose to smoothing $z_b^w(x)$ whenever it has large gradients. But the point to note is that in reality, $z_b^w(x)$ would not have vary large gradients to warrant smoothing. Therefore, for each test case below, given $z_b^w(x)$, we also state the corresponding smooth version (when necessary) used in the implementation of the VCM.*

The outline of this chapter is the following. The test case of previous chapter is repeated for the vertical coupling method in section 6.2 and two more test cases are considered one in each of sections 6.3 and 6.4.

6.2 Test case 1

This test case is the one considered in sections 3.8.3 and 4.8. The simulation of this problem with the full 2D, FBM and horizontal coupling methods have been discussed in chapter 4. Here we discuss only that of the vertical coupling method. For this test case, the channel wall elevation, $z_b^w(x)$ is given as

$$z_b^w(x) = \begin{cases} 2.5, & \text{if } x \leq 12.50, \\ 0.0, & \text{elsewhere.} \end{cases} \tag{6.4}$$

We use the following smoother version of $z_b^w(x)$ for the VCM.

$$z_b^w(x) = \begin{cases} \tanh((x - 10.0)) + 1.0, & \text{if } x < 14.0, \\ 0.0, & \text{if } x \geq 14.0. \end{cases} \tag{6.5}$$

Result of test 1

This problem was simulated using the vertical coupling method with a grid of 193×90 cells in the floodplain and 8 cells in lateral direction inside the channel, making a total of 193×198 non-uniform grid cells, (see table 6.1). Every other simulation data is the same as the other methods. Figure 6.1 displays the free surface elevation for the various methods. It can been seen that the vertical coupling method approximates the full 2D simulation result better than the horizontal coupling method. The accuracy of the vertical coupling method is further illustrated in figure 6.2 which displays the free surface elevation at the six probe points. It can

be seen that the VCM captures the full 2D results very well than the horizontal coupling method, at all the probe points. Again, from figure 6.1, one can see that the VCM recovers 2D flow structure within the channel unlike the HCM and FBM. Again, this 2D flow structure is only within the flooded region, while non-flooded regions remain with 1D solutions. In terms of efficiency, table 6.1 shows that for the

	Grid	No. of time steps	Processor time (in seconds)
Full 2D	193×115	3,669	3,110.31
VCM	193×98	2,615	1,605.31
HCM	193×92	2,616	1,420.40
FBM	193×91	2,592	1,311.26

Table 6.1: Grid cells, simulation times and number of time steps for Test 1. Shows the grid used for each simulation method and the number of time steps and processor time taken to complete the simulation.

this test case, the vertical coupling method has about 48.39% gain in computational time over the full 2D simulation.

6.3 Test case 2 : Channel Flow into Elevated 2D Floodplain

This test case involves the same channel in the previous test case but connected to an elevated floodplain located in the region $10.5 \leq x \leq 16.0$ (see figure 6.3). The channel bed is flat and the floodplain bed is 0.5 meters high. The initial condition is the following:

$$H(x,y,0) = \begin{cases} 1.5, & \text{if } x \leq 8.5, \quad y \geq 1.8, \\ 0.7, & \text{if } x > 8.5, \quad y >= 1.8, \\ 0.2, & \text{if } 10.5 \leq x \leq 16.0, \quad 0 \leq y \leq 1.8, \\ 0.0, & \text{else.} \end{cases} \tag{6.6}$$

$$\bar{u}(x,y,0) = \bar{v}(x,y,0) = 0. \tag{6.7}$$

The boundaries are only open at the sides indicated exit in figure 6.3, others are closed. We use and implement the reflective (no flux) boundary conditions on the closed boundaries and the transmissive (open) boundary condition on the open boundaries.

The manning coefficient for both channel and floodplain is taken as $0.009 s/m^{1/3}$.

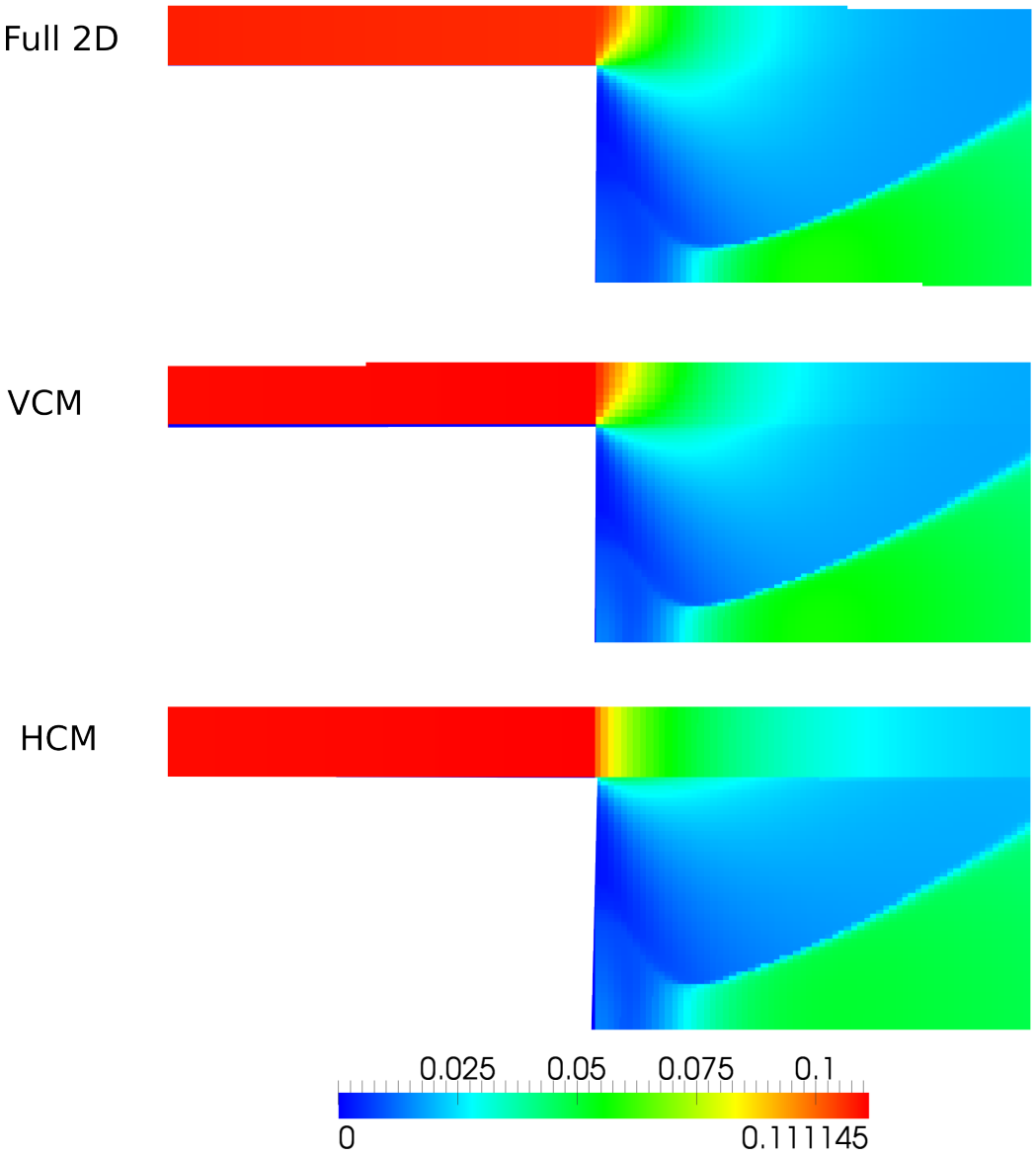


Figure 6.1: Comparison of free surface elevation for the different methods for test case 1 after the last time step.

Just like the previous test case, here nine probe points are identified as indicated in figure 6.3.

To apply the vertical coupling method to this problem, we first realize that from the bottom topography of the channel and bed given above, the channel wall

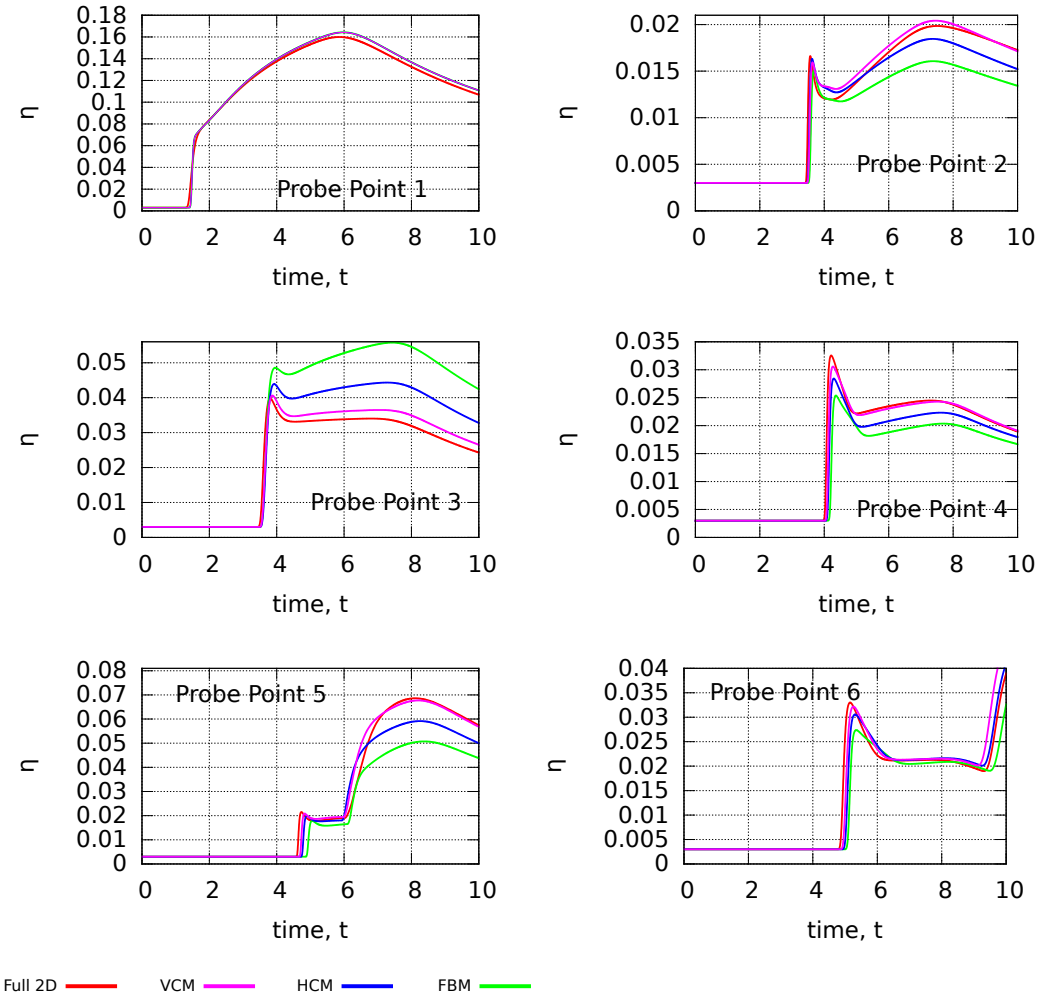


Figure 6.2: Comparison of free surface elevation, η at probe points for test case 1. The locations of the probe points are shown in 3.10.

elevation, $z_b^w(x)$ is given as

$$z_b^w(x) = \begin{cases} 0.5, & \text{if } 10.5 \leq x \leq 16.0, \\ 3.0, & \text{elsewhere.} \end{cases} \tag{6.8}$$

The smoother version used for the VCM is the following

$$z_b^w(x) = \begin{cases} \tanh(0.5(x-4.5)) + 1.5, & \text{if } x < 10.0, \\ 0.5, & \text{if } 10.0 \leq x \leq 16.5, \\ \tanh(x-19.2) + 1.5, & \text{elsewhere.} \end{cases} \tag{6.9}$$

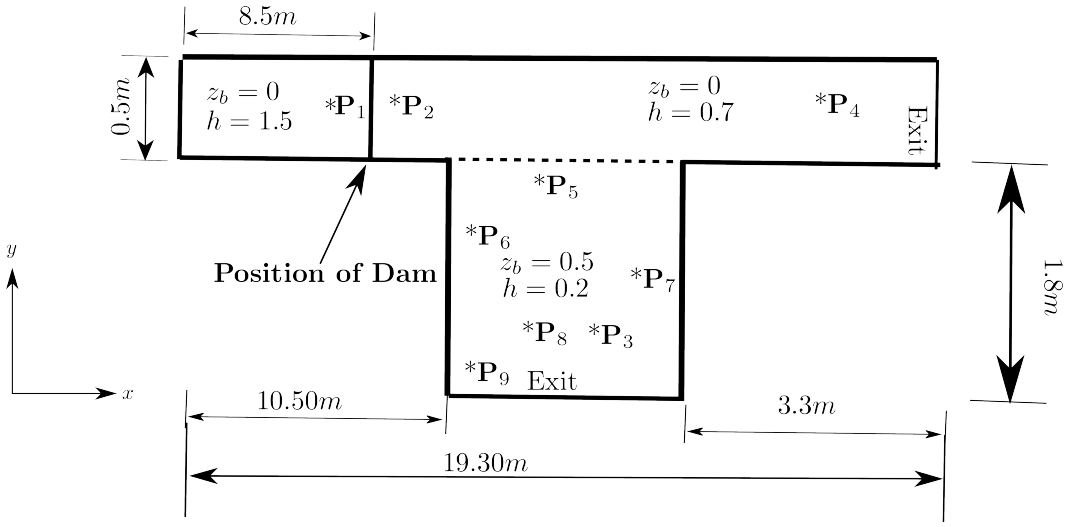


Figure 6.3: Top view of Channel and Floodplain for test case 2 showing the floodplain region in $(x, y) \in [10.5, 16.0] \times [0, 1.8]$ and the channel region in $(x, y) \in [0, 19.3] \times [1.8, 2.3]$.

Result of test 2

The vertical coupling method and the other three methods simulated this problem with their respective grid for the previous problem. This is shown under the grid column in table 6.2. The simulation was run for ten seconds.

Figures 6.4 and 6.5 show the free surface elevation and velocity magnitude for each method. It can be seen that the vertical coupling method provides a better approximation of the full 2D results than the horizontal coupling method which in turn, is more accurate than the FBM. As further validation of this claim, the free surface elevation, the x-component and y-component velocity are plotted for selected probe points in figures 6.6, 6.7 and 6.8 respectively. It can be seen that the vertical coupling method is more accurate than the other methods for all three flow flow quantities, at the reported points and almost throughout the duration of simulation. Also, the horizontal coupling method is more accurate than the FBM at the points for all flow quantities and almost all the time. Again, the two new methods, especially the vertical coupling method, really captures the flow structure of the full 2D simulation. This proves the accuracy of the proposed methods.

For efficiency, we see from column under processing time in table 6.2 that the FBM is very efficient but not very accurate, the vertical coupling method is also efficient compared to the full 2D but not as much as the other two methods, while the horizontal coupling method is both very efficient and also has good accuracy.

	Grid	No. of time steps	Processor time (in seconds)
Full 2D	193×115	4,963	4,100.47
VCM	193×98	3,278	2,811.00
HCM	193×92	3,235	1,710.36
FBM	193×91	3,178	1,555.08

Table 6.2: Grid cells, simulation times and number of time steps : Test 2. Shows the grid used for each simulation method and the number of time steps and processor time taken to complete the simulation.

For this problem, due to the very high accuracy of the vertical coupling method, it is sensible to conclude that it is the best amongst the three coupling methods. Moreover, for a simulation of a very large network of rivers where the 2D region might be very small compared to the entire computational domain, the efficiency of one coupling method over the other might not be very significant, so accuracy becomes the deciding factor. In this regard, the vertical coupling method is the best method of the three considered here.

Finally, figures 6.4 and 6.5 show that the VCM, unlike the other methods, recovers the 2D flow structure within the channel at the flooding regions. Again, the VCM continues to compute 1D solutions at no flooded regions; this demonstrates the self-adaptive nature of the method.

6.4 Test case 3 : Flooding of an initially dry floodplain

The final test case involves the overflowing of a channel onto an initially dry floodplain. Both the channel and the floodplain are located in the $2D$ domain, $[0, 20] \times [0, 4]$. The channel occupies the region, $[0, 20] \times [y_c, 4]$ with flat bottom, $Z_b(x) = 0$, while the floodplain occupies the rest of the domain, $[0, 20] \times [0, y_c]$, where $y_c = 3$. The bottom topography of the entire domain is the following

$$z_b(x, y) = \begin{cases} Z_b(x) = 0, & \text{if } y \geq y_c \\ 0.2 + \frac{z_b^w(x) - 0.2}{y_c} y, & \text{otherwise} \end{cases} \tag{6.10}$$

where $z_b^w(x)$ is the channel wall elevation function defined as

$$z_b^w(x) = \begin{cases} -0.06 \tanh(3(x - 9)) + 0.14, & \text{if } x \leq 10.5 \\ 0.06 \tanh(3(x - 15.5)) + 0.14, & \text{otherwise.} \end{cases} \tag{6.11}$$

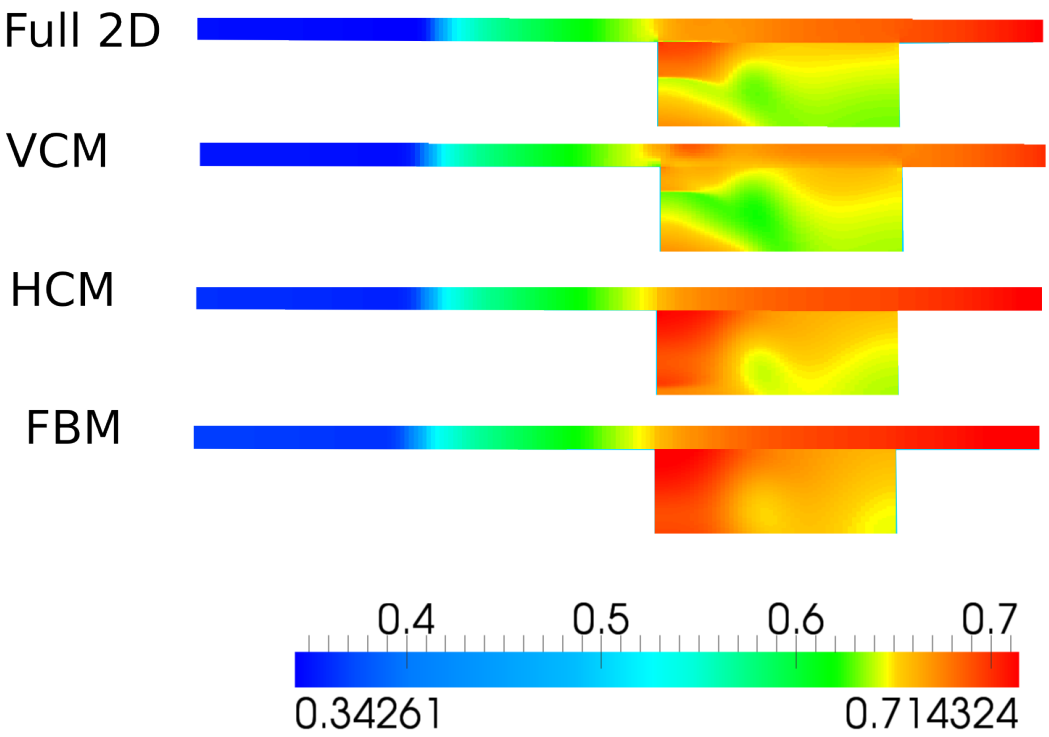


Figure 6.4: Comparison of free surface elevation distribution for the different methods for test case 2 at the last time step.

For this problem, we take $z_b^w(x)$ the way it is. The bed and channel wall elevations are depicted in figure 6.9.

The initial condition consists of stationary water of depth, 0.08 meters in the channel and dry floodplain. The boundary conditions are time-dependent water depth at the left boundary of the channel and zero velocity at the right channel boundary, namely

$$H(0, y, t) = \begin{cases} h_b(t), & \text{if } t \leq 4a, \\ h_b(4a), & \text{if } t > 4a, \end{cases}$$

for $y \geq y_c$.

$$\bar{u}(20, y, t) = 0.0, \text{ for all } t \geq 0, y \geq y_c.$$

where

$$h_b(t) = \eta_0 + r + r \sin\left(\frac{(t-a)\pi}{2a}\right) \tag{6.12}$$

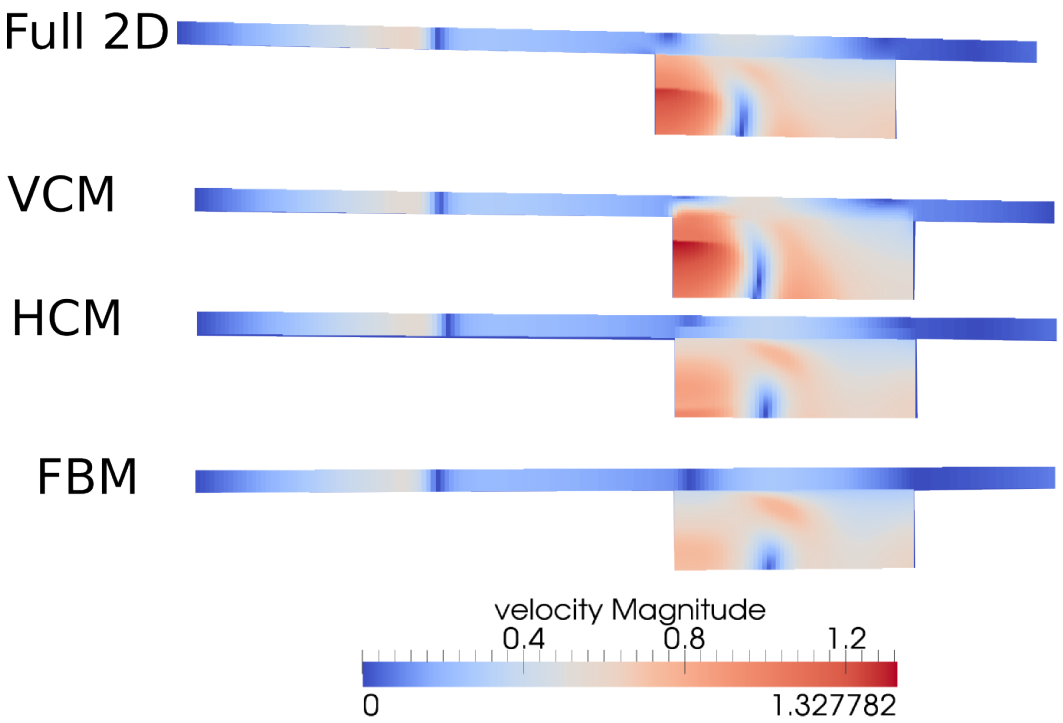


Figure 6.5: Comparison of velocity magnitude for the different methods for test case 2 after the last time step.

where $a = 10$ and $r = 0.025$. $\eta_0 = 0.08$ is a constant initial free-surface elevation inside the channel. The remaining boundaries are closed.

The manning coefficients are the same as used in the previous cases. The following probe points are chosen, $P_1(2.5, 3.5)$, $P_2(4.0, 3.8)$, $P_3(7.0, 3.3)$, $P_4(10.0, 3.4)$, $P_5(11, 3.5)$, $P_6(12, 3.3)$, $P_7(14, 3.4)$, $P_8(16, 3.5)$, $P_9(17.3, 3.5)$, $P_{10}(19, 3.5)$, $P_{11}(12, 2.8)$, $P_{12}(13, 2.8)$, $P_{13}(12, 2.5)$, $P_{14}(12, 2.0)$ and $P_{15}(13.0, 1.0)$.

Discussion of Results of test case 3

Table 6.3 shows the domain discritization for both the channel and floodplain for each method being discussed. As before, all methods use the same grid for the floodplain but different grids for the channel. This problem was simulated for $t = 100$ seconds. We report, in figures 6.10 - 6.13, the results of the simulation after 40 seconds and in figures 6.14 and 6.15, we report the results at selected probe points throughout the duration of the simulation.

As can been seen from the pictures, all the coupling methods provide very good approximation of full 2D simulation results for both the free surface elevation (figure 6.10), the velocity components (figures 6.11 and 6.12) and the velocity mag-

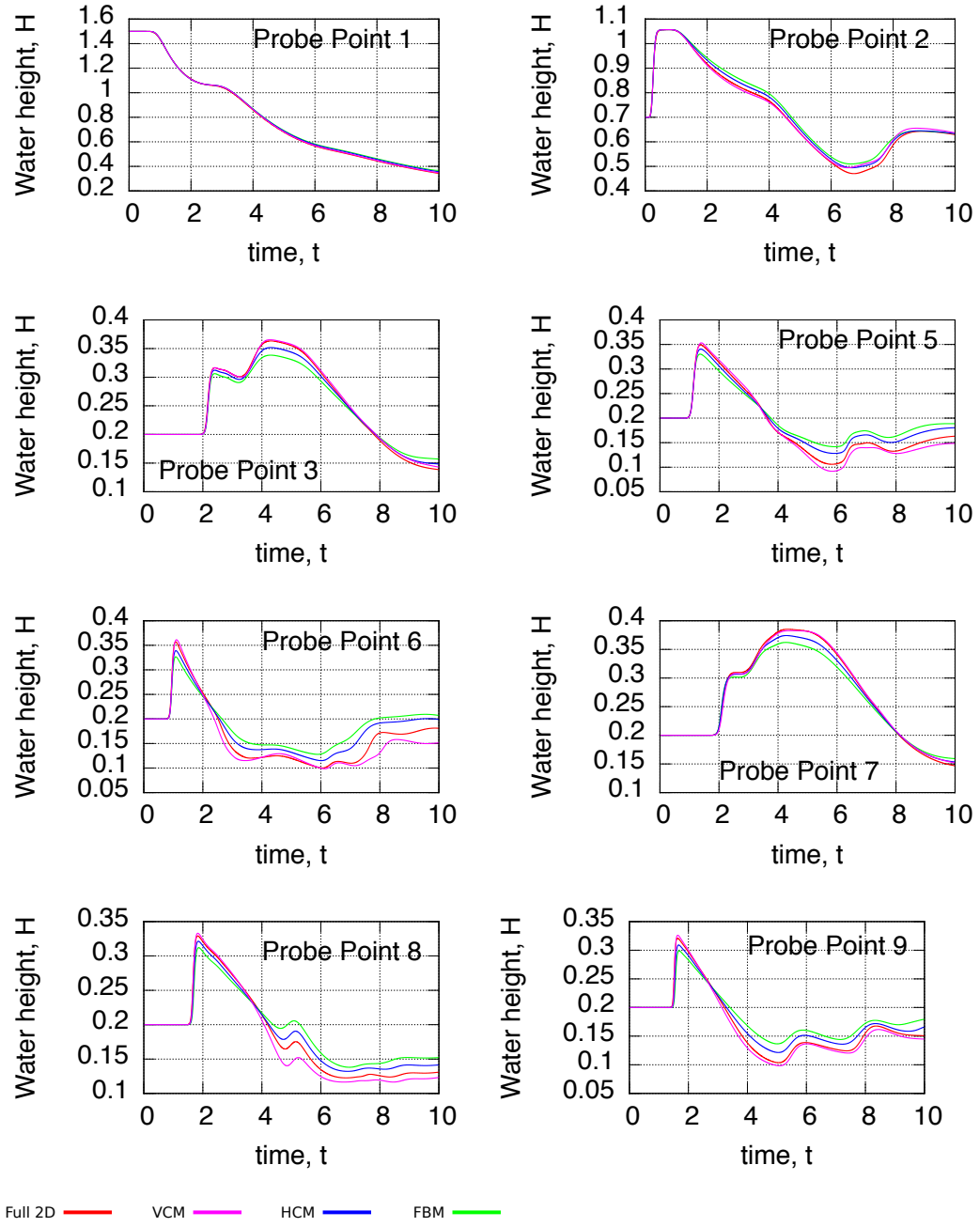


Figure 6.6: Comparison of the time variation of the water height at the probe points for test case 2.

nitude (figure 6.13) for this test case. However, as one can see from these pictures, the VCM provides more smooth variation of flow variables between the channel and the floodplain. That is, all flow variables vary smoothly from the channel to the floodplain. This is not so for the other coupling methods. Again, the lateral varia-

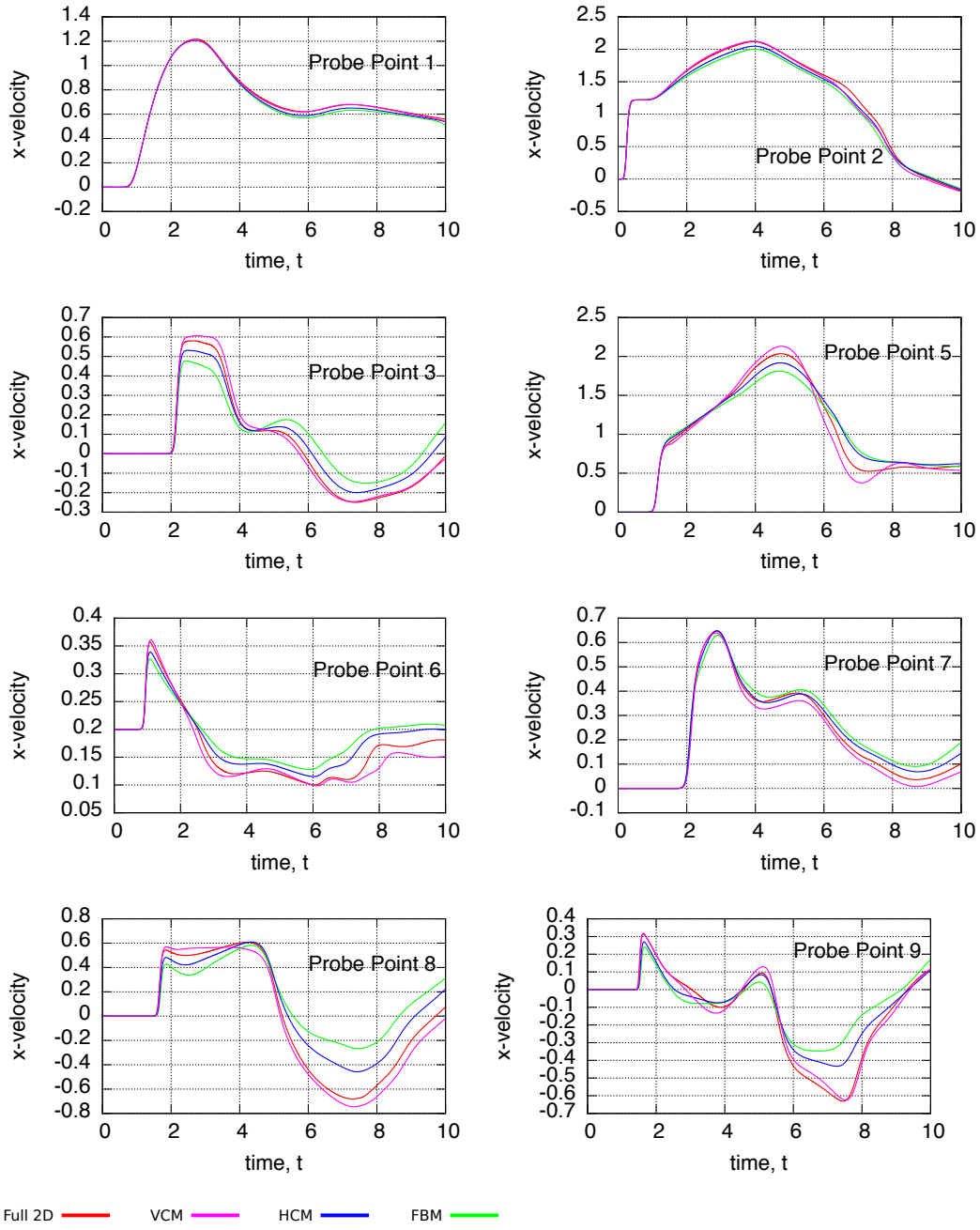


Figure 6.7: Comparison of time variation of the x-velocity component at the probe points for test case 2.

tion of flow quantities within the channel during flooding, is properly captured by the vertical coupling method than the other methods. This demonstrates the ability of the VCM to recover 2D flow structure unlike most existing methods.

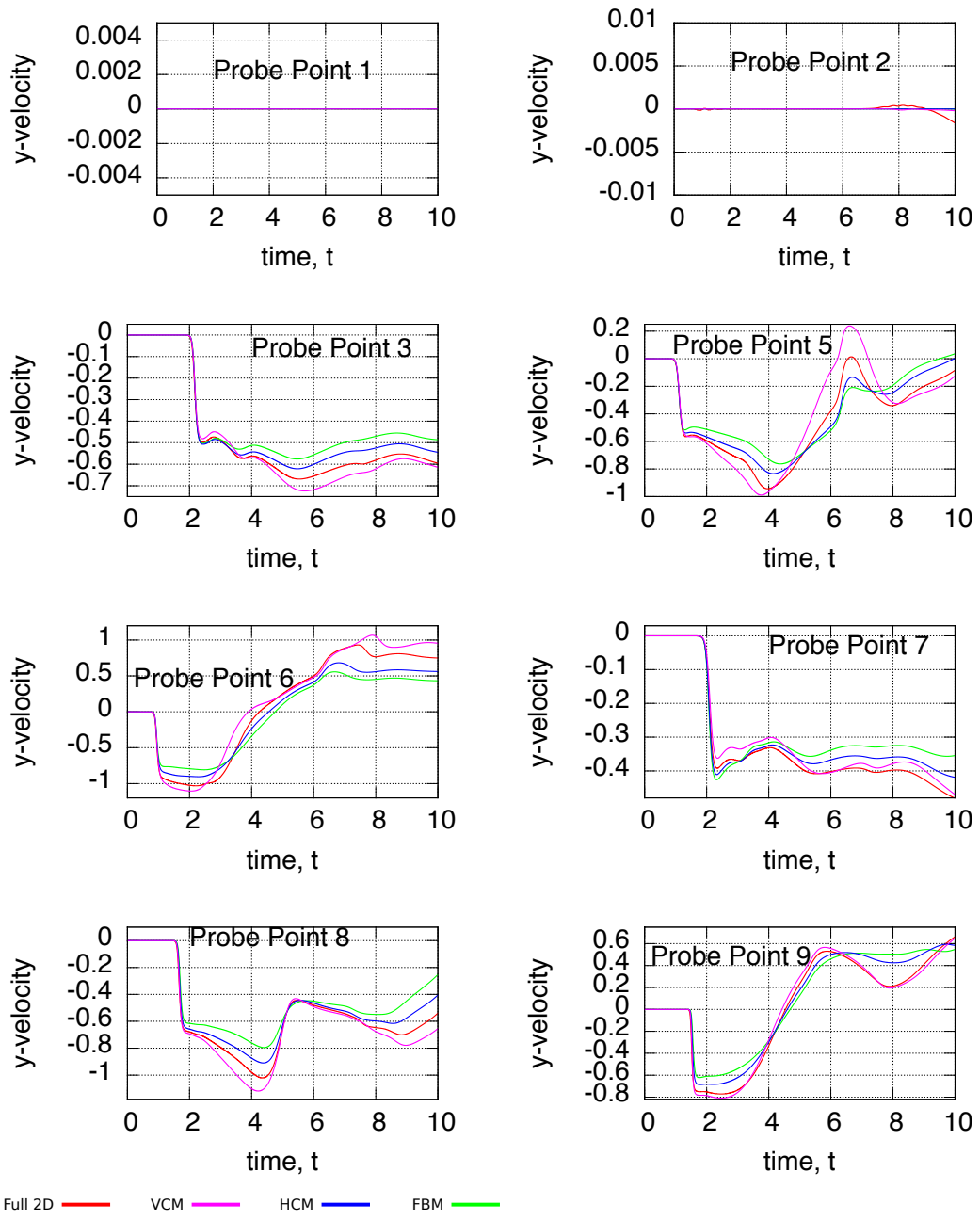
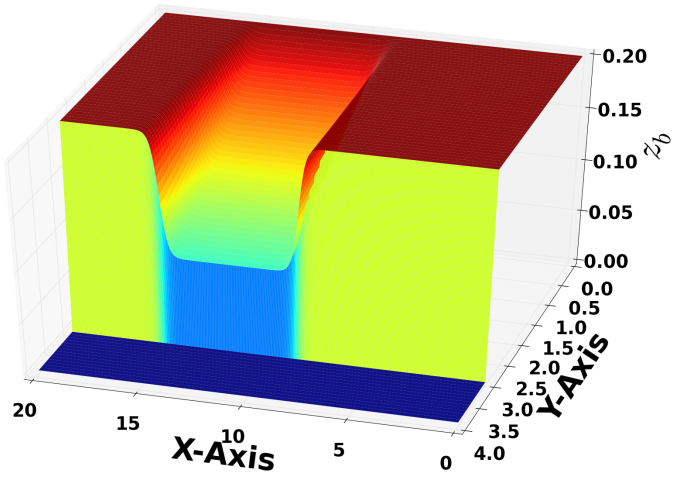


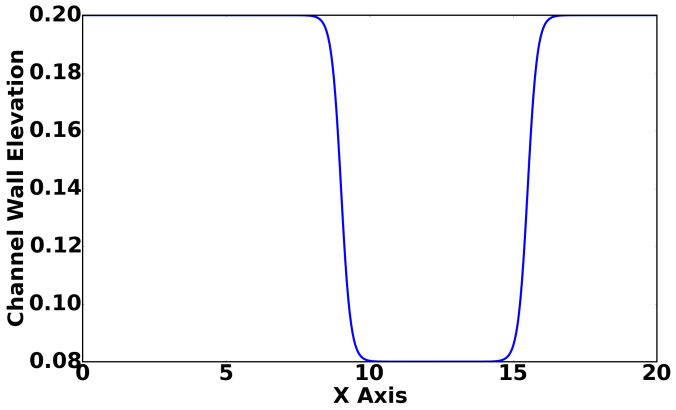
Figure 6.8: Comparison of the time variation of the y-velocity component at probe the points for test case 2.

And in terms of accuracy of y-velocity component, the VCM and HCM provide better approximations than the FBM as can be seen in figure 6.12.

To further understand the results of the simulations, the flow quantities over time at the probe points, $P_1 - P_{15}$ have been explored. Here we report the results



(a) Bottom topography, $z_b(x, y)$ for both the channel and the floodplains.



(b) The channel wall elevation, $z_b^w(x)$.

Figure 6.9: The bottom topography and the channel wall elevation for test 3

at the probe points P_1, P_3, P_5, P_8 and P_{10} which are in the channel and the points, P_{11}, P_{12}, P_{13} and P_{14} in the floodplain. Figures 6.14 and 6.15 show the results for the selected points in the channel and floodplain respectively. In each figure, the left column displays the water depth, the second (middle) column shows the $x-$component of velocity, while the third(right) column shows the $y-$component of velocity.

From figure 6.14, we can see that all the coupling methods provide very good approximation of the results of the full 2D simulations, especially for the water depth

	Channel Grid	Floodplain Grid
Full 2D	600×30	600×90
VCM	600×10	600×90
HCM	600×2	600×90
FBM	600×1	600×90

Table 6.3: Grid cells for test case 3 showing the grids used for the floodplains and the channel for each simulation method.

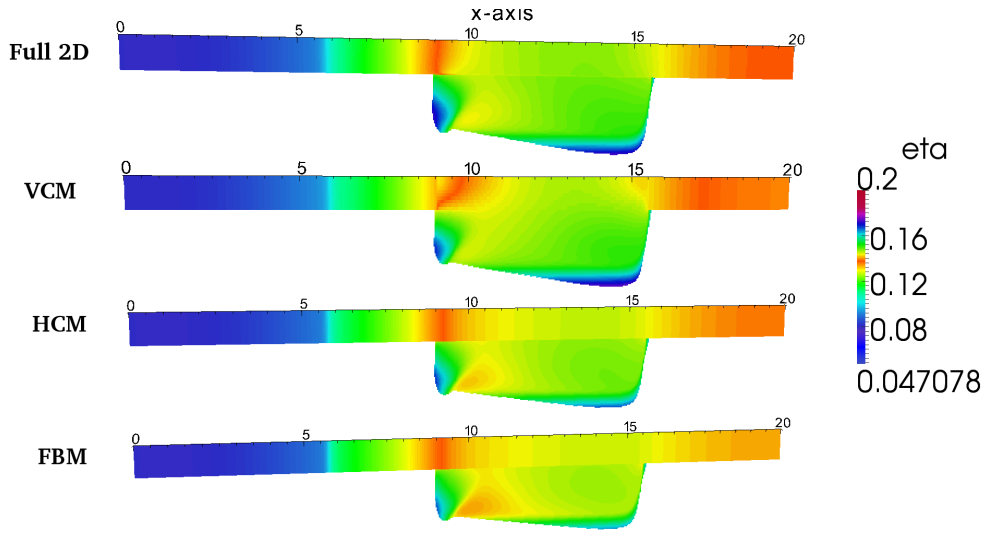


Figure 6.10: Visualisation of free surface elevation after $t = 40$ for test case 3. The x-axis is from left to right, while the y-axis is from the bottom to the top.

and $x-$component of velocity. The $y-$velocity is truly small within the channel and only the VCM and HCM are able to compute its small variations in the channel. This is also true for the probe points not reported here.

From figure 6.15, we also see that for the points in the floodplain, the coupling methods computed very good approximations of results of the full 2D simulation with the HCM and VCM computing more accurate results especially for the $y-$velocity. This figure also verify the no-numerical flooding property of the methods. That is, the floodplain initially remained dry until the time when water height rose above the channel banks. This is the reason why, for all points in the floodplain, the water depth and velocity remained at zero for the first several seconds of the simulation. Another thing to note is that due to the time-dependent boundary condition provided for this problem, water flew onto the floodplain and after sometime the level of water in the channel decreased, so water in the floodplain drained back into the channel leaving the floodplain dry again. The coupling meth-

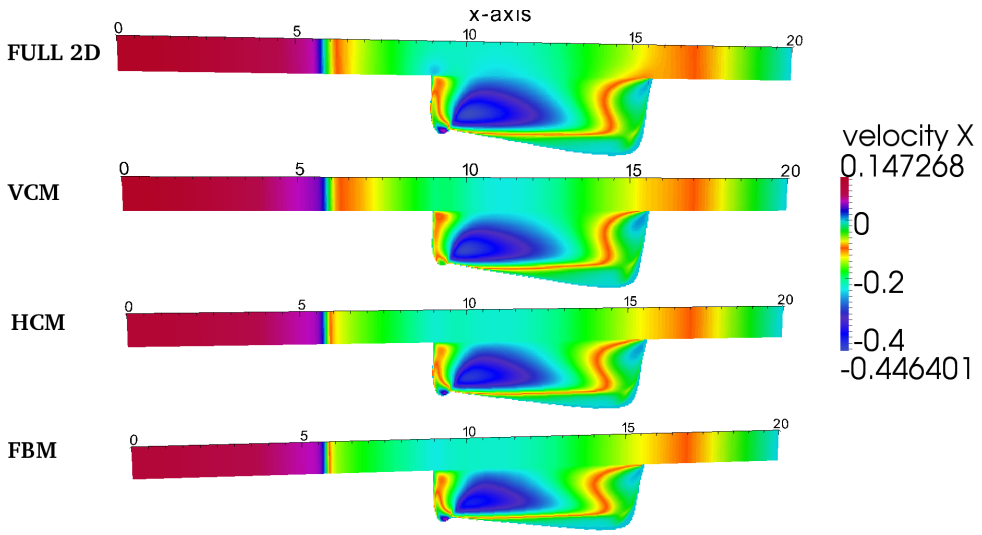


Figure 6.11: Visualisation of x-velocity after $t = 40s$ for test case 3. The x-axis is from left to right, while the y-axis is from the bottom to the top

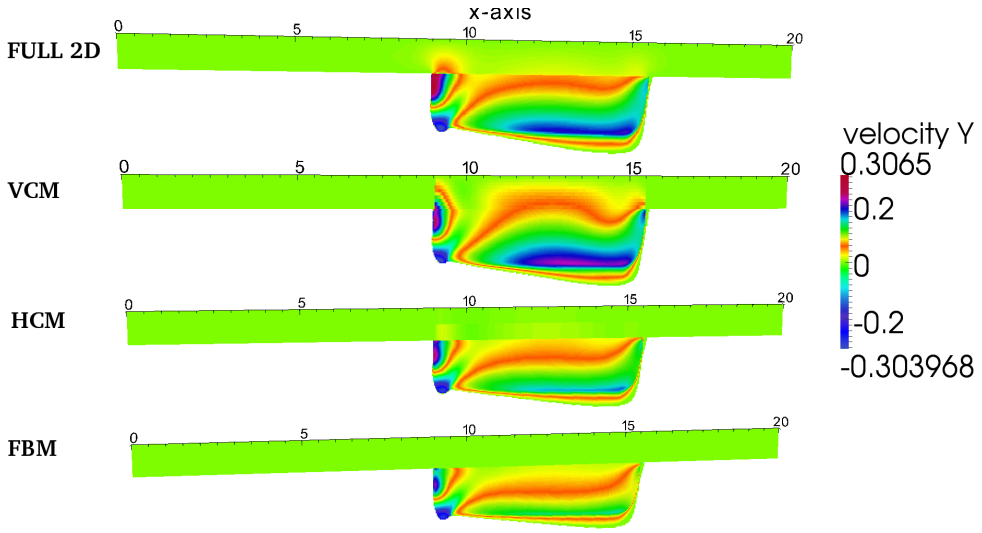


Figure 6.12: Visualisation of y-velocity after $t = 40s$ for test case 3. The x-axis is from left to right, while the y-axis is from the bottom to the top.

ods truly capture this phenomenon as one can see in figure 6.15 where the water depth and velocity return to zero towards the end of the simulation and remain at zero throughout the remaining times of the simulation. This is true for all the points in the floodplain, even those not reported here.

The well-balance property of the methods are also exhibited here. The points in the channel are numbered according to their distance from the left channel boundary. So, P_{10} is further than P_8 which is further than P_7, etc. The boundary condition

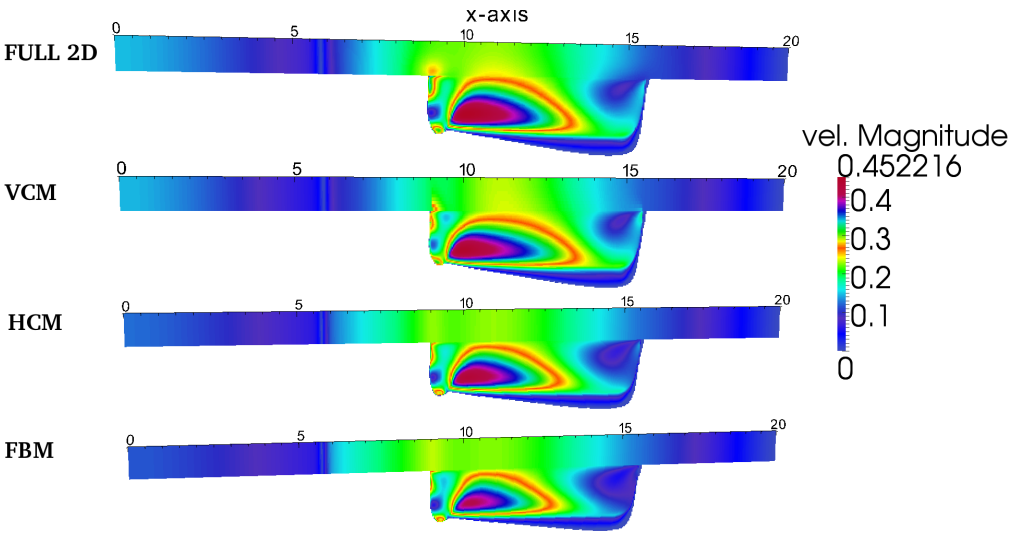


Figure 6.13: Visualisation of velocity magnitude after $t = 40s$ for test case 3. The x-axis is from left to right, while the y-axis is from the bottom to the top

makes water enter into the channel (which is initially at rest) from this left boundary. Therefore, water wave would get to P_1 before P_3 etc. This is also reproduced by the results. From figure 6.15, one can see that the arrival times increase with the distances of the points and that all flow quantities remain at their initial state until the time when the wave arrives. This is true for all the points in the channel and even in the floodplain.

In conclusion, while all the three coupling methods computed results with most of the desirable properties, only the VCM is able to compute 2D flow structure within the channel during flooding. This property sets the VCM apart from other coupling methods.

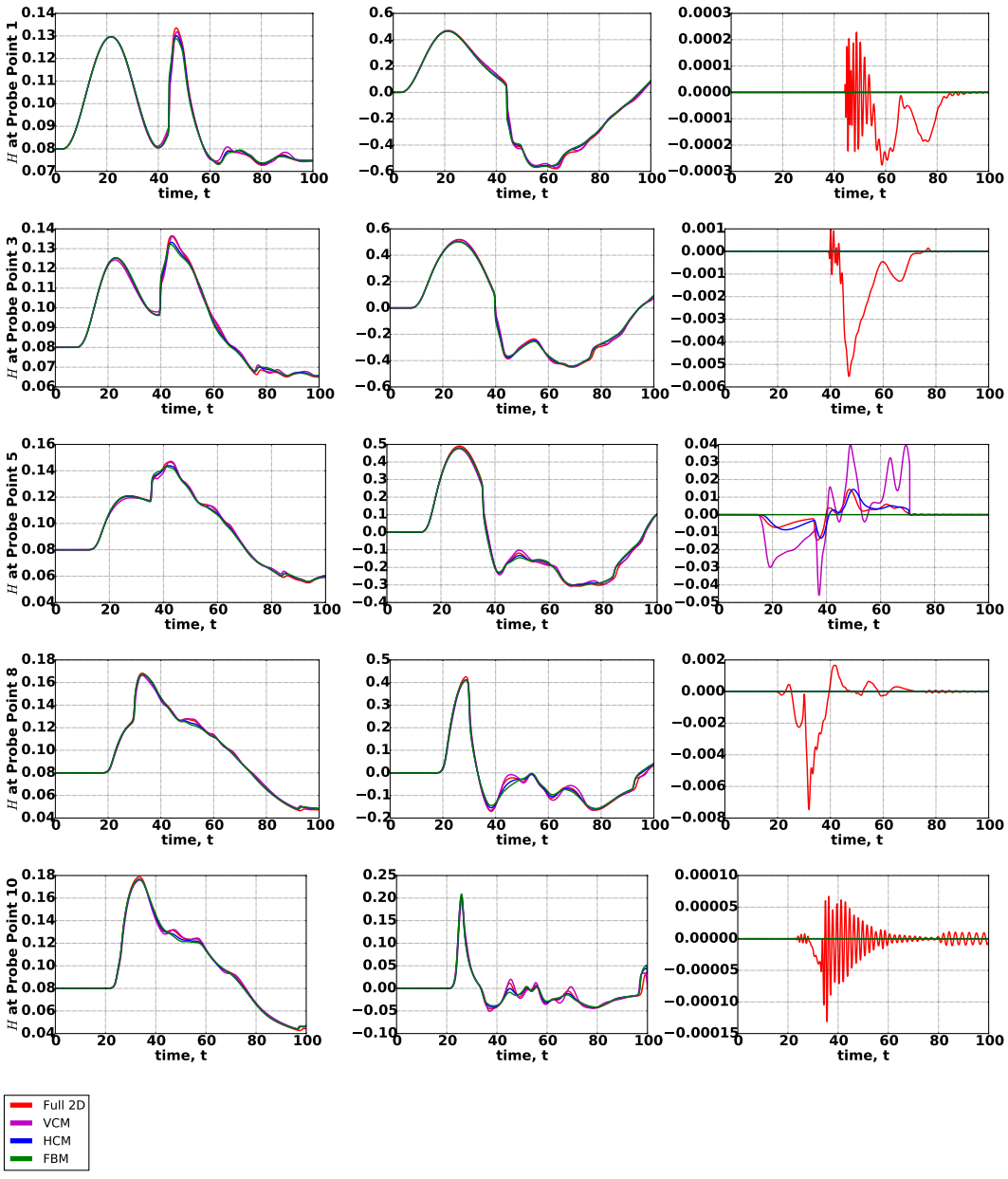


Figure 6.14: Time variation of water depth H (left column), x-velocity component (middle column) and y-velocity component (right column) at selected probe points within the channel for test case 3. Each row corresponds to one probe point which is indicated on the y-axis of the water depth H plot on the left.

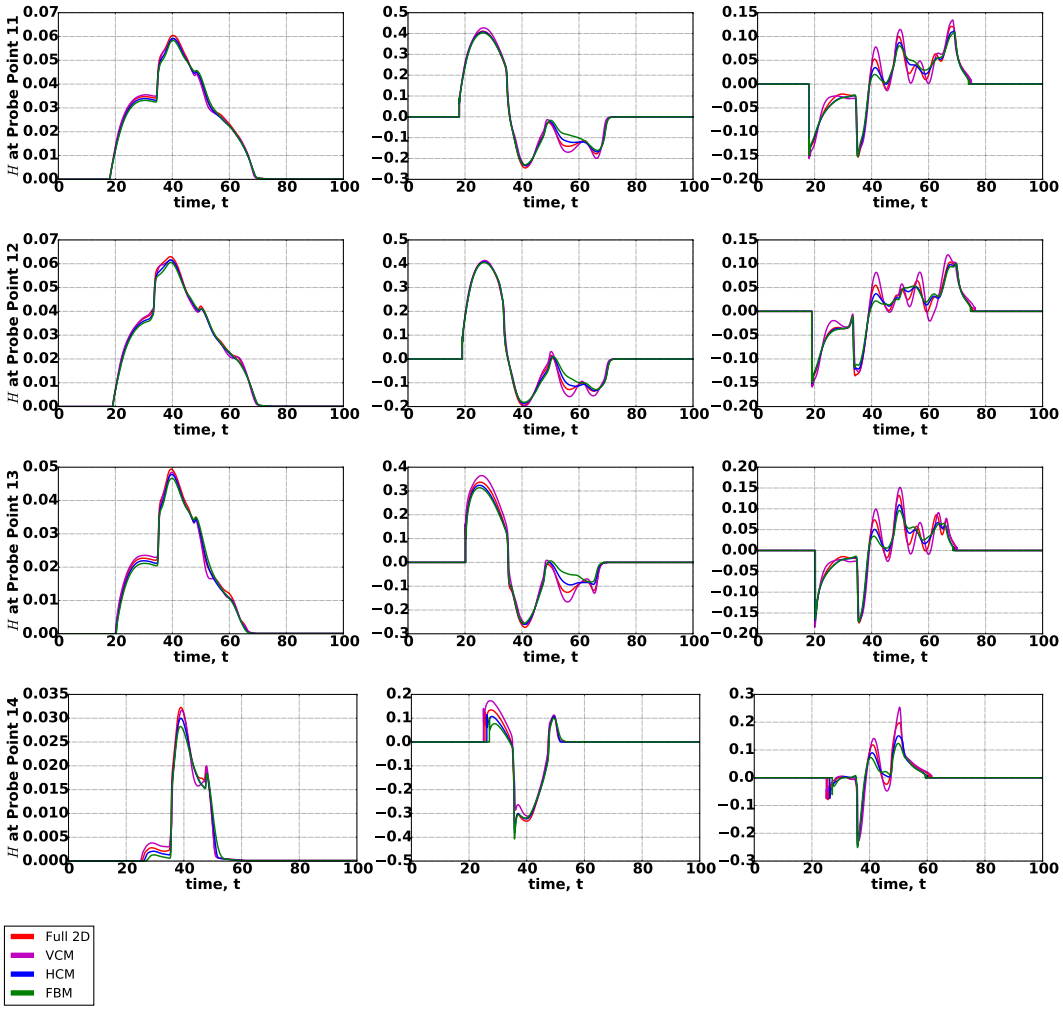


Figure 6.15: Time variation of water depth H (left column), x-velocity component (middle column) and y-velocity component (right column) at the indicated probe points in the floodplain for test case 3. Each row corresponds to one probe point which is indicated on the y-axis of the water depth H plot on the left.

Chapter 7

Summary, Conclusion and Recommendations

7.1 Summary

In this thesis, we have investigated the coupling of 2D and 1D shallow water equations. Apart from the obvious and popular objectives of achieving good accuracy and high efficiency, the key questions considered in this thesis are those about the lateral variation of flow quantities within the channel. Specifically, we asked the following questions:

- Do lateral variations in flow quantities within the channel have any influence on the solution accuracy, in the event of flooding?
- Can a coupling method recover 2D flow structure within the channel during flooding?
- If the answer to (7.1) above is yes, can that be done without solving the full 2D SWE within the channel all the time, including when there is no flooding?
- We also enquired about the possibility of developing a unifying method for a range of existing methods.

To answer the first question, the HCM was formulated in chapter 4 where the lateral discharge was computed within the channel and the result compared with the FBM. The results show that solution was significantly improved by computing this quantity. Hence, we conclude that, truly the lateral variations in flow quantities, though always neglected in most coupling methods, can have a significant influence on the quality of the solution. The HRM is very easy to implement.

To answer the rest questions on a more complete method that allows to recover 2D flow structure within the channel and on a unifying method, the VCM has been proposed in chapter 5. And the VCM answered yes to all these questions. From the test cases discussed in chapter 6, the VCM recovers the 2D flow structure within the channel whenever overflowing. The VCM only activates its two layer algorithm whenever flooding and returns to one layer 1D channel algorithm whenever it does not flood. And as we explained in section 6.1, the VCM is also a unifying method based on the choice of the interface between the two layers. Both frontal and lateral coupling methods can be recovered by this scheme.

With regards to other desirable properties of coupling methods, the results presented in this thesis show that the methods proposed herein, the VCM and HCM compute adequate results, are efficient, well-balanced and preserve the no-numerical flooding. Therefore, the major contribution of this research work are the following

1. Development of a new lateral coupling method, the HCM which is well-balanced, mass conservative, preserve "no-numerical flooding" property and has good performance in terms of accuracy and efficiency. Moreover, the HCM has demonstrated that 2D/1D coupling term can be obtained following the approach proposed by [Marin and Monnier, 2009] but without the need to place any restriction on the channel width variation.

2. Proposed a new approach to model coupling, the VCM which:

   - computes the 2D flow structure if flooding and self-adapts to 1D channel simulation if not flooding;
   - automatically detects flooding regions;
   - does not face the difficulty in computing the channel lateral discharges during flooding;
   - is a unifying method for some existing methods depending on the choice of the location of the layers' interface;
   - is well-balanced, mass conservative, preserves a no-numerical flooding property and performs well in terms of accuracy and efficiency.

Hence, we make the following conclusions.

7.2 Conclusion

From the research reported in this thesis, we therefore make the following conclusions.

1. The lateral variations in flow quantities which are usually neglected in most coupling methods, truly influence the accuracy of the coupling methods.

2. Adequately computing the lateral discharge even with a simple 2D solver, improves the quality of solution. This is exactly what the HCM does.

3. It is possible to recover 2D flow structure in the channel with a coupling method without having to solve 2D problem in the channel all the time. This is what the VCM does.

4. It is possible to develop a single coupling method that is a unifying method and can compute both flow direction and lateral coupling problems. This is the direction which the VCM should be explored further.

5. It is also possible to derive coupling term following the approach of [Marin and Monnier, 2009] without imposing any restriction on the channel width variation.

7.3 Recommendation for further study

The methods proposed in this thesis have shown promising results, however, possible further work are suggested in the following areas.

1. Channel Wall Height function: We observed that the VCM requires moderately smooth variations in the channel wall elevation, $z_b^w(x)$. But we have not completely understood how much smoothness is required on $z_b^w(x)$ in order to get very good results. A further work in understanding the influence of the choice of $z_b^w(x)$ on the results is highly recommended.

2. Intermediate Solution : As mentioned in chapter 5, it is our opinion that different methods could be adopted to solve the sub-layer models, (5.46) and (5.47) with or without their exchange terms. We suggest a further research on alternative means to solve those models.

3. Flow Direction: We have discussed in chapter 6 that the VCM is a family of methods, that is a superset of existing methods depending on the choice of $z_b^w(x)$ but we have not tested the scheme with a frontal coupling code. It would be desirable to implement a frontal coupling method from the literature and compared the results with those of the VCM. A further work is thus suggested in this direction.

4. Unstructured grid: Although, our formulations in this thesis adopt an unstructured grid approach, the test cases considered were for structured grids. We, therefore, suggest further work on testing the methods, especially the VCM for general unstructured grid.

5. Real Data: Another possible work is to apply the methods proposed here for real life cases which might involve real river and real floodplain elevation data and implemented in stuctured or unstructured grid.

6. Sophisticated black-box solvers: Since the methods proposed here do not depend on the existing 1D and 2D solvers, it would be interesting to extend them to other methods like discontinuous Galerkin finite element methods. Also, the methods can be extended to high-order incorporating more sophisticated wetting and drying algorithms.

7. Incorporation into open source code: It would be interesting to also implement these methods in an open source software such as DUNE or OpenFoam.

8. Parallelisation: Writing a parallel code for these methods would also be desirable.

9. Multiphysics: Another aspect of work is to apply this to coupled surface and underground water flow model.

Appendix A

Numerical Flux Functions

A.1 Godunov-Type Finite Volume Schemes

Godunov-type finite volume methods are a class of methods that provide stable and consistent approximation of the numerical fluxes [George, 2006]. The classical Godunov's method [Godunov, 1959] defines a Riemann problem at each cell interface, with initial data corresponding to the cell averages of the bounding cells. That is, to approximate the numerical flux at the interface, $x_{i+1/2}$, the following Riemann problem is defined at $x_{i+1/2}$.

$$\begin{aligned} &\partial_t w + \partial_x f(w) = 0 \\ &w(x, t^n) = \begin{cases} w_i^n \text{ if } x < x_{x+1/2} \\ w_{i+1}^n \text{ if } x > x_{i+1/2} \end{cases} \end{aligned} \tag{A.1}$$

The solution of the Riemann problem, (A.1), which we denote here by $w^n_{i+1/2}(x,t)$, is self-similar (see Toro [2001, 1999]; Leveque [1992, 2002], etc) , that is, $w^n_{i+1/2}(x,t)$ can be written as a function of one variable, namely

$$w^n_{i+1/2}(x,t) = w^n_{i+1/2}\left(\frac{x - x_{i+1/2}}{t}\right) \tag{A.2}$$

Finally, the Godunov's method takes the numerical flux, $\phi^n_{i+1/2}$ to be equal to $f(w^n_{i+1/2}(0))$, that is, the analytical flux function evaluated with the exact solution of the Riemann problem (which in turn is evaluated at $x = x_{i+1/2}$). Then this numerical flux is used to evaluate the scheme, (3.6) for the update value, w_i^{n+1}.

However, the classical Godunov's method is associated with some issues. The Godunov's method requires the exact solution of the Riemann problem, (A.1). For

scalar equations, these solutions might be easily obtained and less expensive also, but for complicated systems, like non-linear systems, the solution process requires iterations for non-linear algebraic systems (for example, see Toro [2001, 1999] for exact solutions of Riemann problems for $1D$ shallow water equations and Euler equations respectively). This makes the idea of solving the Riemann problem, exactly, practically unattractive. This led to further search for alternative methods to approximate the solution of the Riemann problem, (A.1) or directly approximate the numerical fluxes without solving the Riemann problem. These alternative methods are called Approximate Riemann Solvers and are the subject of the next section.

A.2 Approximate Riemann Solvers

While the Godunov's method uses the exact solution of the Riemann problem, (A.1), to compute numerical fluxes, approximate Riemann solvers are methods that use either an approximate solution of the Riemann problem,(A.1) or an exact solution of some linearised version of the Riemann problem, (A.1), to compute numerical fluxes. In this section, we briefly present some of these Riemann solvers. Detailed presentation of Riemann solvers can be found in texts such as [Toro, 1999, 2001; Leveque, 1992, 2002].

A.2.1 Roe Solver

An approximate Riemann solver based on the use of the solution of a linearised Riemann problem is the one proposed in [Roe, 1981], commonly called the Roe solver or Roe flux or Roe method. The idea is to first obtain the analytic Jacobian of the conservation law, (3.1), namely

$$J(w) = \frac{\partial f(w)}{\partial w} \tag{A.3}$$

Assuming that we wish to find the numerical flux across the interface between two cells with approximate cell averages, w_L and w_R, then $J(w)$ is linearised by a constant matrix, $\hat{J}(w_L, w_R)$ using the two states, w_L and w_R. The Roe method then considers the following linearised Riemann problem.

$$\begin{aligned} &\frac{\partial w}{\partial t} + \hat{J}(w_L, w_R)\frac{\partial w}{\partial x} = 0 \\ &w(x,0) = \begin{cases} w_L \text{ if } x < 0 \\ w_R \text{ if } x > 0 \end{cases} \end{aligned} \tag{A.4}$$

In the original method proposed in [Roe, 1981], the constant matrix, $\hat{J}$ is explicitly found and it is required to satisfy the following properties.

Hyperbolicity : $\hat{J}(w_L, w_R)$ should have real eigenvalues and complete set of linearly independent eigenvectors.

Consistency : $\hat{J}(w, w) = J(w)$

Conservation : $f(w_R) - f(w_L) = \hat{J}(w_R - w_L)$

The matrix, $\hat{J}$ is called Roe matrix when it satisfies all the above properties. The construction of Roe matrix for Euler's equation is given in [Roe, 1981], see also chapter 11.2.2 in [Toro, 1999]. However, there is no unique choice of this matrix for a particular problem [Pelanti et al., 2001; MacDonald, 1996] and the process of constructing it is quiet complicated. Furthermore, as we shall see shortly, the numerical flux function by Roe method does not explicitly require the Roe matrix, but only requires the eigenvalues and eigenvectors. Consequently, [Roe and Pike, 1985] proposed another approach which avoids the explicit construction of the Roe matrix, instead one computes some average, called Roe average, $\hat{w}$ of w_L and w_R with which the needed eigenvalues, eigenvectors and wave strengths of $\hat{J}$ are obtained by directly evaluating those of the analytical Jacobian, $J(w)$ with $w = \hat{w}$. This alternative approach is the one we follow in this thesis and it is briefly explained below.

Given that $\lambda_k(w), \mathbf{e}_k(w), k = 1, ..., M$ are the eigenvalues and eigenvectors respectively, of the analytic Jacobian, $J(w)$. Then $\hat{\lambda}_k = \hat{\lambda}_k(w_L, w_R) := \lambda_k(\hat{w})$ and $\hat{\mathbf{e}}_k = \hat{\mathbf{e}}_k(w_L, w_R) := \mathbf{e}_k(\hat{w})$ are the needed k-th , $k = 1, 2, ..., M$, eigenvalues and eigenvectors respectively in Roe-Pike approach. Where $\hat{w}$ is a Roe average which needs to be computed depending on the problem. Then with the solution of (A.4), it can be shown (see chapter 11 in [Toro, 1999]) that the numerical flux function of Roe method is given by

$$\phi^{Roe}(w_L, w_R) = \frac{f(w_L) + f(w_R)}{2} - \frac{1}{2}\sum_{k=1}^{M} \hat{\alpha}_k |\hat{\lambda}_k| \hat{\mathbf{e}}_k \tag{A.5}$$

where $\hat{\alpha}_k = \hat{\alpha}_k(w_L, w_R)$ is the k-th wave strength, which is the coefficient of the k-th eigenvector, in the eigenvector expansion of the jump, $w_R - w_L$, namely

$$w_R - w_L = \sum_{k=1}^{M} \hat{\alpha}_k \hat{\mathbf{e}}_k \tag{A.6}$$

(see [Toro, 1999]). As an illustration, the Roe solver for the 1D shallow water equations with transverse velocity is detailed below.

Given the 1D system,

$$\partial_t w + \partial_x F_1(w) = 0 \tag{A.7}$$

where $w = (h, q_x, q_y)^T$ and $F_1(w) = (q_x, \frac{q_x^2}{h} + \frac{g}{2}h^2, \frac{q_x q_y}{h})^T$. The Jacobian matrix of the system is

$$J(w) = \begin{pmatrix} 0 & 1 & 0 \\ c^2 - u^2 & 2u & 0 \\ -uv & v & u \end{pmatrix} \tag{A.8}$$

where $u = q_x/h, v = q_y/h, c = \sqrt{gh}$. The eigenvalues are

$$\lambda_1(w) = u - c, \quad \lambda_2(w) = u \text{ and } \lambda_3(w) = u + c \tag{A.9}$$

and the eigenvectors are

$$\mathbf{e}_1(w) = \begin{pmatrix} 1 \\ u - c \\ v \end{pmatrix}, \quad \mathbf{e}_1(w) = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \text{ and } \mathbf{e}_1(w) = \begin{pmatrix} 1 \\ u + c \\ v \end{pmatrix} \tag{A.10}$$

see chapter 3.3 in [Toro, 2001] for details.

To compute the numerical flux at the interface between two cells with cell averages, w_L and w_R using the Roe-Pike approach, the Roe average $\hat{w} = (\hat{h}, \hat{h}\hat{u}, \hat{h}\hat{v})^T$ is given by [Glaister, 1988] with

$$\hat{u} = \frac{u_L\sqrt{h_L} + u_R\sqrt{h_R}}{\sqrt{h_L} + \sqrt{h_R}}, \quad \hat{v} = \frac{v_L\sqrt{h_L} + v_R\sqrt{h_R}}{\sqrt{h_L} + \sqrt{h_R}}, \quad \hat{h} = \sqrt{h_L h_R}, \quad \hat{c} = \sqrt{g\hat{h}} \tag{A.11}$$

Hence the required eigenvalues and eigenvectors of the Roe matrix, $\hat{J}(w) := J(\hat{w})$ are the following.

$$\hat{\lambda}_1 := \lambda_1(\hat{w}) = \hat{u} - \hat{c}, \quad \hat{\lambda}_2 := \lambda_2(\hat{w}) = \hat{u}, \quad \hat{\lambda}_3 := \lambda_3(\hat{w}) = \hat{u} + \hat{c} \tag{A.12}$$

and

$$\hat{\mathbf{e}}_1 := \mathbf{e}_1(\hat{w}) = \begin{pmatrix} 1 \\ \hat{u} - \hat{c} \\ \hat{v} \end{pmatrix}, \ \hat{\mathbf{e}}_2 := \mathbf{e}_2(\hat{w}) = \begin{pmatrix} 0 \\ 0 \\ \hat{u} \end{pmatrix}, \ \hat{\mathbf{e}}_3 := \mathbf{e}_3(\hat{w}) = \begin{pmatrix} 1 \\ \hat{u} + \hat{c} \\ \hat{v} \end{pmatrix} \quad \text{(A.13)}$$

The wave strengths are obtained by solving the system, (A.6) alongside $\Delta hu = \hat{u}\Delta h + \hat{h}\Delta u, \Delta hv = \hat{v}\Delta h + \hat{h}\Delta v$ (see [Toro, 1999]). This gives

$$\hat{\alpha}_1 = \frac{1}{2}\left[\Delta h - \frac{\hat{h}}{\hat{c}}\Delta u\right], \quad \hat{\alpha}_2 = \hat{h}\Delta v, \quad \hat{\alpha}_3 = \frac{1}{2}\left[\Delta h + \frac{\hat{h}}{\hat{c}}\Delta u\right] \quad \text{(A.14)}$$

where $\Delta s = s_R - s_L$ for any quantity, s.

Using equations (A.11)-(A.14), the numerical flux is computed using the Roe solver, (A.5). In the following section, we discuss a well-known flaw of the Roe method presented above, and also point out one of the available approaches for circumventing this flaw.

Entropy Fix

Before we discuss the entropy fix of Roe method, let us first recall the following definition, see [Toro, 2001].

Definition A.2.1 (Genuinely non-linear fields)**.** *The $\lambda_k, k \in \{1, .., M\}$ characteristic field is said to be genuinely non-linear if the following equation holds*

$$\nabla\lambda_k(w).\boldsymbol{e}(w) \neq 0 \quad \forall w \in \mathbb{R}^3 \quad \text{(A.15)}$$

The Roe method uses the solution of a linearised Riemann problems whose wave speeds (eigenvalues) are all constants. Rarefaction waves are associated with genuinely non-linear fields [Toro, 2001] hence the solution of the linearised Riemann problem in Roe method does not contain rarefaction waves, but only discontinuities. The problem of the Roe method occurs when the true solution of the non-linear Riemann problem, (A.1) contains a transonic rarefaction in the k-th field, $k \in \{1, .., M\}$, that is when the characteristic speed changes from negative to positive across the k-th eigenvalue, namely

$$\lambda_k^L < 0 < \lambda_k^R \quad \text{(A.16)}$$

where λ_k^L, λ_k^R are the characteristic speeds to the left and right respectively, of the k-th eigenvalue. Even in this case, the Roe method would still approximate the

solution with a single discontinuity with speed, λ_k. Now, if this approximation were true, then the entropy condition requires that

$$\lambda_k^L > \lambda_k > \lambda_k^R \tag{A.17}$$

that is, characteristics must run into the discontinuity [Toro, 1999, 2001]. Clearly, Roe approximation using a discontinuity, when the true solution satisfies (A.16), violates the entropy condition (A.17). Therefore, the Roe method would converge to entropy-violating weak solution. It is thus important to recognise when transonic rarefactions occur and correct the Roe method appropriately. This process of correcting the Roe method is, in the literature, referred to as entropy fix (see [Leveque, 1992; Toro, 1999] for example).

A lot of entropy fixes have been proposed for the Roe method, like [Harten and Hyman, 1983]. A review on entropy fixes is given in [Pelanti et al., 2001], see also [Roberts, 2013; Toro, 1999; Leveque, 1992]. Here, we briefly present the entropy fix due to [Harten and Hyman, 1983] as presented in [Pelanti et al., 2001].

Given that we have obtained all the eigenvalues, eigenvectors and wave strengths associated with the Roe average vector, then the solution of the linearised Riemann problem (A.4) is given by

$$w(x,t) = w_L + \sum_{j=1}^{k^*} \hat{\alpha}_j \hat{\mathbf{e}}_j \tag{A.18}$$

where k^* is the index of the largest eigenvalue less than x/t, that is,

$$\hat{\lambda}_{k^*} = \max_{\lambda_k < \frac{x}{t}} \hat{\lambda}_k \tag{A.19}$$

see [Toro, 1999; Leveque, 2002].

Then the entropy fix of Harten and Hyman [Harten and Hyman, 1983] is the following. For each $\lambda_k, k = 1, ..., M$ which is a genuinely non-linear field (see definition A.2.1), take the following steps
i) Compute the states $w_{k,L}, w_{k,R}$ to the left and right of the k-th elementary wave travelling at speed $\hat{\lambda}_k$, namely

$$w_{k,L} = w_L + \sum_{j=1}^{k-1} \hat{\alpha}_j \hat{\mathbf{e}}_j \text{ and } w_{k,R} = w_{k,L} + \hat{\alpha}_k \hat{\mathbf{e}}_k \tag{A.20}$$

(see [Leveque, 1992; Pelanti et al., 2001]).

ii) Compute characteristic speeds associated to these two states, namely

$$\lambda_k^L := \lambda_k(w_{k,L}) \text{ and } \lambda_k^R := \lambda_k(w_{k,R}) \tag{A.21}$$

iii) Investigate the presence of transonic rarefaction by checking if (A.16) is satisfied, then modify the corresponding eigenvalue in the Roe solver, (A.5) as follows.

$$\hat{\lambda}_k^{fixed} = \begin{cases} \frac{(\lambda_k^L+\lambda_k^R)\hat{\lambda}_k - 2\lambda_k^L\lambda_k^R}{\lambda_k^R-\lambda_k^L}, & \text{if } \lambda_k^L < 0 < \lambda_k^R \\ |\hat{\lambda}_k|, & \text{else} \end{cases} \tag{A.22}$$

where $\hat{\lambda}_k^{fixed}$ is the modified eigenvalue. Note that it is not changed if there is no transonic rarefaction.
iv) Finally, the Roe solver is redefined with this modified eigenvalues, namely

$$\phi^{Roe}(u_L, u_R) = \frac{f(u_L)+f(u_R)}{2} - \frac{1}{2}\sum_{k=1}^{M} \hat{\alpha}_k \hat{\lambda}_k^{fixed} \hat{\mathbf{e}}_k \tag{A.23}$$

Therefore, the Roe solver, (A.5) with entropy fix of [Harten and Hyman, 1983], is (A.23) where $\hat{\lambda}_k^{fixed}$ is given in (A.22). See [Pelanti et al., 2001] for more details.

A.2.2 HLL Approximate Riemann Solver

Another approximate Riemann solver is the due to [Harten et al., 1983] which is popularly called the HLL solver. The HLL method directly finds the numerical flux without solving any linear or non-linear system like the Roe and Godonuv methods. Details of the method can be found in [Toro, 1999, 2001], here the flux function is just written down. The HLL solver is given by

$$\phi^{hll}(w_L, w_R) = \begin{cases} f(w_L), & \text{if } s_L \geq 0 \\ f^* := \frac{s_R f(w_L) - s_L f(w_R) + s_L s_R (w_R - w_L)}{s_R - s_L}, & \text{if } s_L \leq 0 \leq s_R \\ f(w_R), & \text{if } s_R \leq 0 \end{cases} \tag{A.24}$$

where s_L and s_R are estimates of the smallest and largest wave speeds in the solution of the Riemann problem (A.1) [Toro, 2001]. There are several choices for s_L, s_R [Toro, 2001, 1999]. They are given in [Toro, 2001] for the 1D shallow water equations,

(2.70) as

$$s_L = u_L - c_L q_L, \quad s_R = u_R + c_R q_R \tag{A.25}$$

where q_p (with $p = L, R$) is given defined as

$$\begin{aligned} q_p &= \begin{cases} \sqrt{\frac{(h_* + h_p)h_*}{2h_p^2}} & \text{if } h_* > h_p \\ 1 & \text{if } h_* \leq h_p, \end{cases} \\ u_p &= \left(\frac{q_x}{h}\right)_p, c_p = \sqrt{g h_p}. \end{aligned} \tag{A.26}$$

and

$$h_* = \frac{1}{g}\left[\frac{1}{2}(c_L + c_R) + \frac{1}{4}(u_L - u_R)\right]^2 \tag{A.27}$$

Alternative definitions for the wave speeds are those given in [Bouchut, 2007], for a general conservation law, (3.1), namely

$$s_L = \min_k\{\lambda_k(w_L), \lambda_k(w_R)\}, \quad s_R = \max_k\{\lambda_k(w_L), \lambda_k(w_R)\} \tag{A.28}$$

where $\lambda_k, k = 1, ..., M$, are the eigenvalues of the Jacobian matrix of the system.

A.2.3 HLLC Approximate Riemann Solver

Recall that the HLL solver accounts for only the smallest and largest wave speeds in the solution of the Riemann problem. This might be fine for a system of two equations like the 1D shallow water equations, but not generally correct for larger systems since there might be significant influence of intermediate waves [Toro et al., 1994]. This HLLC was therefore proposed in [Toro et al., 1994] to account for the influence of the missing intermediate waves in the HLL solver [Toro, 2001, 1999]. The HLLC solver for the 1D shallow water equation with transverse velocity, equation (A.7), is given in [Toro, 2001] as.

$$\phi^{hllc}(w_L, w_R) = \begin{cases} F_1(w_L), & \text{if } s_L \geq 0 \\ F_1^{*L} := F_1(w_L) + s_L(w_{*L} - w_L), & \text{if } s_L \leq 0 \leq s_* \\ F_1^{*R} := F_1(w_R) + s_R(w_{*R} - w_R), & \text{if } s_* \leq 0 \leq s_R \\ F_1(w_R), & \text{if } s_R \leq 0 \end{cases} \tag{A.29}$$

where

$$w_{*p} = h_p\left(\frac{s_p - u_p}{s_p - s_*}\right)\begin{pmatrix}1\\ s_*\\ v_p\end{pmatrix}, \quad p = L, K \tag{A.30}$$

$$s_* = \frac{s_L h_R(u_R - s_R) - s_R h_L(u_L - s_L)}{h_R(u_R - s_R) - h_L(u_L - s_L)} \tag{A.31}$$

An alternative way to express the HLLC solver, is to express it as a function of the HLL solver [Fernandez-Nieto et al., 2008; Toro, 2001], namely

$$\phi^{hllc}(w_L, w_R) = \begin{pmatrix}[\phi^{hll}]_1\\ [\phi^{hll}]_2\\ [\phi^{hll}]_1 v_*\end{pmatrix} \tag{A.32}$$

where

$$v_* = \begin{cases} v_L & \text{if } s_* \geq 0 \\ v_R & \text{if } s_* < 0, \end{cases} \tag{A.33}$$

$[\phi^{hll}]_k$, $(k = 1, 2)$, is the k-th component of the HLL flux in (A.24), s_* is given in (A.31), see [Fernandez-Nieto et al., 2008] and $v_p = \left(\frac{q_y}{h}\right)_p, p = L, R$ is the transverse velocity.

A.3 Centered Numerical Flux Solvers

Another family of methods to approximate numerical flux, ϕ is the centred-type method which is of the form.

$$\phi(w_L, w_R) = \frac{1}{2}\Big(F_1(w_L) + F_1(w_R)\Big) - \frac{1}{2}Q(w_L, w_R)\Big(w_R - w_L\Big) \tag{A.34}$$

where $Q(w_L, w_R)$ is a numerical viscosity whose definition determines the type of scheme. For instance,
i) Lax-Friedrich's scheme :

$$Q(w_L, w_R) = \frac{\Delta x}{\Delta t} \tag{A.35}$$

ii) Modified Lax-Friedrich's :

$$Q(w_L, w_R) = c\frac{\Delta x}{\Delta t}, \quad c = CFL \in (0, 1] \tag{A.36}$$

iii) Rosanov's :

$$Q_{i+1/2} = Q(w_i^n, w_{i+1}^n) := \max_{1 \le k \le 3} (|(\lambda_k)_{i+1/2}|) \tag{A.37}$$

iv) Lax-Wendroff scheme :

$$Q_{i+1/2} = Q(w_i^n, w_{i+1}^n) := \frac{\Delta t}{\Delta x} J_{i+1/2}^2 \tag{A.38}$$

v) FORCE and GFORCE - Convex combination of Lax-Friedrich's and Lax-Wedroff's scheme :

$$\begin{aligned} &Q_{i+1/2} = Q(w_i^n, w_{i+1}^n) := (1-\omega)\frac{\Delta x}{\Delta t}\mathbf{I} + \omega\frac{\Delta t}{\Delta x} J_{i+1/2}^2 \\ &\text{FORCE: } \omega = 0.5, \\ &\text{GFORCE: } \omega = \frac{1}{1+CFL}, \quad CFL \in (0, 1] \end{aligned} \tag{A.39}$$

In all the schemes above, $\mathbf{I}$ is a 3-by-3 identity matrix, $J_{i+1/2}$ is some average (like Roe average) of Jacobian matrix associated with F_1 while $(\lambda_k)_{i+1/2}$ is the k-th eigenvalue of $J_{i+1/2}$ [Toro, 2001; Fernandez-Nieto and Narbona, 2013]. Note that the Jacobian matrix, eigenvalues and eigenvectors are same as those of system, (A.7) as presented in section A.2.1.

Bibliography

O. Agbonkhese, E. G. Agbonkhese, E. O. Aka, J. Joe-Abaya, M. Ocholi, and A. Adenkule. Flood menace in nigeria: impacts, remedial and management strategies. *Civil and Environmental Research*, 6:32–40, 2014.

E. Aldrighetti. *Computational hydraulic techniques for the Saint Venant equations in arbitrarily shaped geometry*. PhD thesis, Universitá Degli Studi di Trento, Italy, 2007.

E. Audusse and M. Bristeau. A well-balanced positivity preserving second-order scheme for shallow water flows on unstructured meshes. *Journal of Computational Physics*, 206(1):311–333, 2005.

E. Audusse, F. Bouchut, M.O. Bristeau, R. Klein, and B. Perthame. A fast and stable well-balanced scheme with hydrostatic reconstruction for shallow water flows. *SIAM Journal Scientific Computing*, 25:2050–2065, 2004.

E. Audusse, M. Bristeau, B. Perthame, and J. Sainte-Marie. A multilayer saint-venant system with mass exchanges for shallow water flows. derivation and numerical validation. *ESAIM: Mathematical Modelling and Numerical Analysis*, 45 (01):169–200, 2011.

A. Bermudez and M. E. Vazquez. Upwind methods for hyperbolic conservation laws with source terms. *Computers & Fluids*, 23(8):1049–1071, 1994.

E. Bladé, M. Gómez, and J. Dolz. Quasi-two dimensional modelling of flood routing in rivers and flood plains by means of storage cells. In *Modelling of Flood Propagation Over Initially Dry Areas*, pages 156–170. ASCE, 1994.

E. Bladé, M. Gómez-Valentín, J. Dolz, JL Aragón-Hernández, G. Corestein, and M. Sánchez-Juny. Integration of 1d and 2d finite volume schemes for computations of water flow in natural channels. *Advances in Water Resources*, 42:17–29, 2012.

F. Bouchut. *Nonlinear stability of finite Volume Methods for hyperbolic conservation laws: And Well-Balanced schemes for sources.* Springer Science & Business Media, 2004.

F. Bouchut. *Efficient Numerical Finite Volume Schemes for Shallow Water Models.* Elsevier, 2007.

F. Bouchut and T. M. de Luna. Semi-discrete entropy satisfying approximate riemann solvers. the case of the suliciu relaxation approximation. *Journal of Scientific Computing*, 41(3):483–509, 2009.

F. Bouchut, J. Le Sommer, and V. Zeitlin. Frontal geostrophic adjustment and nonlinearwave phenomena in one dimensional rotating shallow water. *Part*, 2: 35–63, 2004.

A. Bressan. Lecture notes on hyperbolic conservation laws. 2009.

M. Bristeau and B. Coussin. *Boundary conditions for the shallow water equations solved by kinetic schemes.* PhD thesis, INRIA, 2001.

J. Burguete and P. García-Navarro. Efficient construction of high-resolution tvd conservative schemes for equations with source terms: application to shallow water flows. *International Journal for Numerical Methods in Fluids*, 37(2):209–248, 2001.

Y. Chen, Z. Wang, Z. Liu, and D. Zhu. 1d-2d coupled numerical model for shallow-water flows. *Journal of Hydraulic Engineering.*, 138:122–132, 2012.

J. A. Cunge, F. M. Holly, and A. Verwey. Practical aspects of computational river hydraulics. 1980.

A. de Boer. Comparison of two numerical schemes for solving the 1d shallow water equations. Master's thesis, 2003.

A. B. de Saint-Venant. Théorie du mouvement non-permanent des eaux avec application aux crues des riviéres at á l'introduction des marée dans leur lits. *Acad. Sci. Comptes Rendus Paris*, 73:237–240, 1871.

A. Decoene, L. Bonaventura, E. Miglio, and F. Saleri. Asymptotic derivation of the section-averaged shallow water equations for natural river hydraulics. *Mathematical Models and Methods in Applied Sciences*, 19(03):387–417, 2009.

O. Delestre, C. Lucas, P. Ksinant, F. Darboux, C. Laguerre, T. N. Vo, F. James, and S. Cordier. Swashes: a compilation of shallow water analytic solutions for hydraulic and environmental studies. *International Journal for Numerical Methods in Fluids*, 72(3):269–300, 2013.

S. Doocy, A. Daniels, S. Murray, and T. D. Kirsch. The human impact of floods: a historical review of events 1980-2009 and systematic. 2013.

D. R. Durran. *Numerical methods for fluid dynamics: With applications to geophysics*, volume 32. Springer Science & Business Media, 2010.

L. C. Evans. *Partial Differential Equations*. American Mathematical Society, 2010.

E. D. Fernandez-Nieto and G. R. Narbona. *Lecture Notes on Finite volume methods in environmental fluid mechanics*. CIMPA Research school, Saint-Louis, Snegal, 2013.

E. .D Fernandez-Nieto, D. Bresch, and J. Monnier. A consistent intermediate wave speed for a well-balanced hllc solver. *Comptes Rendus Mathematique*, 346(13): 795–800, 2008.

E. D. Fernandez-Nieto, J. Marin, and J. Monnier. Coupling superposed 1d and 2d shallow-water models: Source terms in finite volume schemes. *Computers & Fluids*, 39(6):1070–1082, 2010.

D. L. George. *Finite volume methods and adaptive refinement for tsunami propagation and inundation*. PhD thesis, 2006.

R. Ghostine, I. Hoteit, J. Vazquez, A. Terfous, A. Ghenaim, and R. Mose. Comparison between a coupled 1d-2d model and a fully 2d model for supercritical flow simulation in crossroads. *Journal of Hydraulic Research*, 53(2):274–281, 2015.

P. Glaister. Approximate riemann solutions of the shallow water equations. *Journal of Hydraulic Research*, 26(3):293–306, 1988.

S. K. Godunov. A difference method for numerical calculation of discontinuous solutions of the equations of hydrodynamics. *Matematicheskii Sbornik*, 89(3): 271–306, 1959.

N. Goutal, M. Parisot, and F. Zaoui. A 2d reconstruction for the transverse coupling of shallow water models. *International Journal for Numerical Methods in Fluids*, 75(11):775–799, 2014.

J. M. Greenberg and A. Leroux. A well-balanced scheme for the numerical processing of source terms in hyperbolic equations. *SIAM Journal on Numerical Analysis*, 33(1):1–16, 1996.

A. Harten. High resolution schemes for hyperbolic conservation laws. *Journal of computational physics*, 49(3):357–393, 1983.

A. Harten and J. Hyman. Self adjusting grid methods for one-dimensional hyperbolic conservation laws. *Journal of computational Physics*, 50(2):235–269, 1983.

A. Harten, P. D. Lax, and B. Van Leer. On upstream differencing and godunov-type schemes for hyperbolic conservation laws. *SIAM review*, 25(1):35–61, 1983.

J. Hou, F. Simons, M. Mahgoub, and R. Hinkelmann. A robust well-balanced model on unstructured grids for shallow water flows with wetting and drying over complex topography. *Computer Methods in Applied Mechanics and Engineering*, 257:126–149, 2013.

J. Hou, Q. Liang, H. Zhang, and R. Hinkelmann. An efficient unstructured muscl scheme for solving the 2d shallow water equations. *Environmental Modelling & Software*, 66:131–152, 2015.

T. Y. Hou and P. G. Le Floch. Why nonconservative schemes converge to wrong solutions: Error analysis. *Math. Comput.*, 62(206):497–530, April 1994. ISSN 0025-5718. doi: 10.2307/2153520. URL `http://dx.doi.org/10.2307/2153520`.

N. M. Hunter, P. D. Bates, S. Neelz, G. Pender, I. Villanueva, N. G. Wright, D. Liang, R. A. Falconer, B. Lin, and S. Waller. Benchmarking 2d hydraulic models for urban flood simulations. In *Proceedings of the Institution of Civil Engineers: Water Management*, volume 161, pages 13–30. Thomas Telford (ICE publishing), 2008.

A. Jackson. Flooding, 2016. URL `https://geographyas.info/rivers/flooding/`.

T. Katsaounis, B. Perthame, and C. Simeoni. Upwinding sources at interfaces in conservation laws. *Applied mathematics letters*, 17(3):309–316, 2004.

D. B. Kharat. *Practical aspects of integrated 1D2D flood modelling of urban floodplains using LiDAR topography data*. PhD thesis, Heriot-Watt University, 2009.

D. Lannes. *Water Waves Problem. Mathematical Analysis and Asymptotics*. AMS, 2013.

P. Lax and B. Wendroff. Systems of conservation laws. *Communications on Pure and Applied mathematics*, 13(2):217–237, 1960.

P. D. Lax. *Hyperbolic Systems of Conservation Laws and mathematical Theory of Shock Waves*. CBMS-NSF Regional Conference Series in Applied Mathematics, 1973.

R. J. Leveque. *Numerical Methods for Conservation Laws*. Birkhäuser verlag, 1992.

R. J. Leveque. *Finite volume methods for hyperbolic problems*, volume 31. Cambridge university press, 2002.

Q. Liang. Flood simulation using a well-balanced shallow flow model. *Journal of hydraulic engineering*, 136(9):669–675, 2010.

I. MacDonald. *Analysis and computation of steady open channel flow*. PhD thesis, 1996.

A. Mangeney-Castelnau, F. Bouchut, J. P. Vilotte, E. Lajeunesse, A. Aubertin, and M. Pirulli. On the use of saint venant equations to simulate the spreading of a granular mass. *Journal of Geophysical Research: Solid Earth*, 110(B9), 2005.

J. Marin and J. Monnier. Superposition of local zoom models and simultaneous calibration for 1d-2d shallow water flows. *Mathematics and Computers in Simulation*, 80(3):547–560, 2009.

E. Mignot, A. Paquier, and S. Haider. Modeling floods in a dense urban area using 2d shallow water equations. *Journal of Hydrology*, 327(1):186–199, 2006.

M. Morales-Hernández. *Efficient Explicit Finite Volume Schemes for the shallow water equations with solute transport*. PhD thesis, Universidad Zaragoza, 2014.

M. Morales-Hernández, P. García-Navarro, and J. Murillo. A large time step 1d upwind explicit scheme (cfl¿ 1): Application to shallow water equations. *Journal of Computational Physics*, 231(19):6532–6557, 2012.

M. Morales-Hernández, P. García-Navarro, J. Burguete, and P. Brufau. A conservative strategy to couple 1d and 2d models for shallow water flow simulation. *Computers & Fluids*, 81:26–44, 2013.

M. Morales-Hernández, G. Petaccia, P. Brufau, and P. García-Navarro. Conservative 1d–2d coupled numerical strategies applied to river flooding: The tiber (rome). *Applied Mathematical Modelling*, 40(3):2087–2105, 2016.

S. Mungkasi. *A study of well-balanced finite volume methods and refinement indicators for the shallow water equations.* PhD thesis, The Australian National University, 2012.

S. Noelle, N. Pankratz, G. Puppo, and J. R. Natvig. Well-balanced finite volume schemes of arbitrary order of accuracy for shallow water flows. *Journal of Computational Physics*, 213(2):474–499, 2006.

M. Pelanti, L. Quartapelle, and L. Vigevano. A review of entropy fixes as applied to roes linearization. *Teaching material of the Aerospace and Aeronautics Department of Politecnico di Milano*, 2001.

G. Pender and S. Néelz. Use of computer models of flood inundation to facilitate communication in flood risk management. *Environmental Hazards*, 7(2):106–114, 2007.

M. Pitt. The pitt review: Learning lessons from the 2007 floods. *United Kingdom Government*, 2007.

G. Puppo. Numerical entropy production on shocks and smooth transitions. *Journal of scientific computing*, 17(1-4):263–271, 2002.

G. Puppo. Numerical entropy production for central schemes. *SIAM Journal on Scientific Computing*, 25(4):1382–1415, 2004.

M. Renardy and R. C. Rogers. *An Introduction to Partial Differential Equations.* Springer, 2004.

S. G. Roberts. Numerical solution of conservation laws applied to the shallow water wave equations. 2013.

P. L. Roe. Approximate riemann solvers, parameter vectors, and difference schemes. *Journal of computational physics*, 43(2):357–372, 1981.

P. L. Roe and J. Pike. Efficient construction and utilisation of approximate riemann solutions. In *Proc. of the sixth int'l. symposium on Computing methods in applied sciences and engineering, VI*, pages 499–518. North-Holland Publishing Co., 1985.

L. A. Rossman, R.E. Dickson, T. Schade, C. Chan, E.H Burgess, and L.A. Huber. Swmm5: The usepa's newest tool for urban drainage analysis. In *Procedings of 10th International Conference on Urban Drainage*, Copenhgen, Denmark, 2005.

A. Settari and D. Walters. Advances in coupled geomechanical and reservoir modeling with applications to reservoir compaction. *Spe Journal*, 6(03):334–342, 2001.

S. D. Seyoun, Z. Vojinovic, R. K. Price, and S. Weesakul. Coupled 1d and noninertia 2d flood inundation model for simulation of urban flooding. *Journal of Hydraulic Engineering*, 138:23–34, 2012.

A. B. Smith and R. W. Katz. Us billion-dollar weather and climate disasters: data sources, trends, accuracy and biases. *Natural hazards*, 67(2):387–410, 2013.

J. J. Stoker. *Water Waves. The Mathematical Theory with Applications*. Interscience, 1957.

R. Szymkiewicz. *Numerical Modeling in Open Channel Hydraulics*. Water Science and Technology, 2010.

J. Tanguy. *Fluvial and Maritime Currentology Models*. Wiley-ISTE, 2010.

P. Tawari-Fufeyin, M. Paul, and A. O. Godleads. Some aspects of a historic flooding in nigeria and its effects on some niger-delta communities. *American Journal of Water Resources*, 3(1):7–16, 2015.

W. C. Thacker. Some exact solutions to the nonlinear shallow-water equations. *J. Fluid Mech*, 107:499–508, 1981.

E. F. Toro. *Riemann solvers and numerical methods for fluid dynamics: a practical introduction*. Springer Science & Business Media, 1999.

E. F. Toro. *Shock Capturing Methods For Free-surface Flows*. Wiley, 2001.

E. F. Toro, M. Spruce, and W. Speares. Restoration of the contact surface in the hll-riemann solver. *Shock waves*, 4(1):25–34, 1994.

M. E. Vázquez-Cendón. Improved treatment of source terms in upwind schemes for the shallow water equations in channels with irregular geometry. *Journal of Computational Physics*, 148(2):497–526, 1999.

M. E. Vázquez-Cendón. *Solving Hyperbolic Equations with Finite Volume Methods*, volume 90. Springer, 2015.

T. Viseu, A. Franco, and A. B. de Almeida. Numerical and computational results of the 2-d biplan model. In *4th Meeting of the Working Group on Dam-Break Modelling (1st CADAM Meeting)*, 1999.

O. Von Estorff and C. Hagen. Iterative coupling of fem and bem in 3d transient elastodynamics. *Engineering Analysis with Boundary Elements*, 30(7):611–622, 2006.

Y. Wang. Numerical improvements for large-scale flood simulation. 2011.

M. Zhang, Y. Xu, Z. Hao, and Y. Qioa. Intragating 1d and 2d hydrodynamic, sediment transport model for dam-break flow using finite volume method. *Science China-Physics, Mechanics and Astronomy*, 4:774–783, 2014.